



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TEMA: “ATERRIZAJE AUTOMÁTICO DE UN VEHÍCULO
AÉREO NO TRIPULADO BASADO EN SEGUIMIENTO DE
PUNTOS DE INTERÉS PARA SUPERFICIES MÓVILES”**

AUTOR:

SALCEDO PEÑA, VINICIO STALIN

DIRECTOR: DR. AGUILAR CASTILLO, WILBERT G.

SANGOLQUÍ

2018



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICADO

Certifico que el trabajo de titulación “**ATERRIZAJE AUTOMÁTICO DE UN VEHÍCULO AÉREO NO TRIPULADO BASADO EN SEGUIMIENTO DE PUNTOS DE INTERÉS PARA SUPERFICIES MÓVILES**” ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas – ESPE, por lo tanto me permito acreditarlo y autorizar al señor **VINICIO STALIN SALCEDO PEÑA** para que lo sustente públicamente.

Sangolquí, 26 de Febrero de 2018

A handwritten signature in blue ink is positioned above a horizontal line. The signature is stylized and appears to read 'W. Aguilar'.

Dr. Wilbert G. Aguilar

DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, **VINICIO STALIN SALCEDO PEÑA**, con cédula de identidad N° 1724458151, declaro que este trabajo de titulación “**ATERRIZAJE AUTOMÁTICO DE UN VEHÍCULO AÉREO NO TRIPULADO BASADO EN SEGUIMIENTO DE PUNTOS DE INTERÉS PARA SUPERFICIES MÓVILES**”, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 26 de Febrero de 2018

Vinicio Stalin Salcedo Peña

C.C. 1724458151



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo Vinicio Stalin Salcedo Peña, autorizo a la Universidad de las Fuerzas Armadas Espe publicar en la biblioteca virtual de la institución el presente trabajo de titulación “**ATERRIZAJE AUTOMÁTICO DE UN VEHÍCULO AÉREO NO TRIPULADO BASADO EN SEGUIMIENTO DE PUNTOS DE INTERÉS PARA SUPERFICIES MÓVILES**”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 26 de Febrero de 2018

Vinicio Stalin Salcedo Peña

C.C. 1724458151

DEDICATORIA

Dedico este trabajo a la persona más importante en mi vida, mi Madre, Mariana Peña ya que, con su lucha incansable, su perseverancia, su total entrega y amor hacia sus hijos nos ha sabido llevar de la mano a lo largo de nuestras vidas entregándonos todo lo que es suyo para que podamos salir adelante, inculcándonos sus valores éticos y guiándonos a pesar de cualquier circunstancia por el camino del bien. Este trabajo y toda mi carrera no habrían sido posibles sin su apoyo y entrega incondicionales siendo mi principal fuente de inspiración y mi más grande ejemplo de sabiduría y fortaleza.

También dedico este trabajo a mi Hermano, José Luis que durante toda mi vida ha sido el amigo máspreciado que tengo y una persona admirable por sus valores, sus virtudes y su espíritu soñador y valiente, ha sido una fuente de inspiración constante y ejemplo a seguir para mí.

A mi novia Gabriela y a mis amigos con quienes he compartido momentos de felicidad y tristeza que con el pasar del tiempo se han convertido en personas a las que respeto y aprecio muchísimo entre ellos: Danny, Juan-k, Jhon, Flaco, Taka, Daniel y todos los demás que me supieron apoyar en los momentos duros y brindarme su sabiduría.

Vinicio Stalin Salcedo Peña.

AGRADECIMIENTO

Agradezco en primer lugar a mi Madre por ayudarme todo el tiempo y por brindarme su incondicional afecto y soporte en todo sentido durante mi carrera y mi vida, agradezco que me haya brindado una educación de calidad y me haya inculcado sus valores que me caracterizan como persona.

Agradezco a mi hermano por todos los consejos que me ha dado, los momentos de felicidad y el apoyo que siempre me ha brindado. Le agradezco por ser una persona a quien respeto muchísimo y por ayudarme a ser alguien mejor y no rendirme ante las adversidades.

Agradezco a mi novia Gabriela ya que a pesar de haber llegado hace poco a mi vida me ha apoyado incondicionalmente a cada instante, gracias a su cariño que me ha llevado a querer ser siempre mejor y sentirme complementado, le agradezco por su insistencia y esfuerzo pues gracias a su apoyo logré emprender el viaje a Brasil para defender la publicación realizada durante la tesis y le agradezco por acompañarme siempre y saber comprender mi personalidad.

Agradezco a mi tutor de tesis el Doctor Wilbert de quien durante el transcurso de este proyecto he aprendido a nivel profesional y personal y me ha sabido dar las bases para seguir mi camino como investigador, le agradezco por su apoyo y guía durante el proceso y la posteridad.

Finalmente agradezco a mis amigos y compañeros que me apoyan en cada proceso, su apoyo es pilar fundamental para continuar en el camino de la excelencia.

Vinicio Stalin Salcedo Peña.

ÍNDICE DE CONTENIDOS

DEDICATORIA.....	v
AGRADECIMIENTOS.....	vi
ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xii
CAPITULO I.....	1
GENERALIDADES.....	1
1.1 Planteamiento del Problema	1
1.2 Antecedentes.....	3
1.3 Justificación e importancia	6
1.4 Alcance del proyecto	7
1.5 Objetivos.....	8
1.5.1 Objetivo General.....	8
1.5.2 Objetivos Específicos	8
1.6 Descripción del proyecto de investigación	9
CAPITULO II	12
MARCO TEÓRICO	12
2.1 Conceptos generales de los Vehículos Aéreos Multirotores no Tripulados	12
2.1.1 Parámetros físicos	12
2.1.2 Dinámica del movimiento.....	13
2.2 Captura de Imágenes de la cámara	16
2.2.1 Representación de una imagen	16
2.2.2 Formación de una imagen.....	17
2.2.3 Modelo de Pin-hole de la cámara	18
2.3 Caracterización de las imágenes	21
2.3.1 Métodos de caracterización de imágenes	22
2.3.2 Proceso de caracterización de una imagen	24

2.3.3 Algoritmos de detección y descripción de puntos de interés.....	32
2.4 Transformaciones Geométricas	39
2.4.1 Transformación de similitud.....	40
2.4.2 Transformación afín.....	40
2.4.3 Transformación de Perspectiva.....	42
2.5 Control de un UAV.....	43
CAPITULO III.....	45
GENERALIDADES DEL SISTEMA.....	45
3.1 Hardware del Sistema	45
3.1.1 Vehículo aéreo no tripulado.....	45
3.1.2 Plataforma terrestre.....	49
3.1.3 Estación en tierra	51
3.2 Software del sistema	51
3.2.1 Comunicación usando ROS.....	52
3.2.2 Uso del controlador de ROS para Bebop Drone.....	53
3.3 Extracción de imágenes	56
3.3.1 Parámetros de imagen.....	57
3.3.2 Subscriptor de imágenes	58
3.4 Control de movimiento del dron.....	60
3.4.1 Clase para manipulación del controlador “Bebop_ autonomy”	62
3.4.2 Comandos de detención del dron.....	64
3.5 Mensajes de estado del dron	65
CAPITULO IV	68
DETECCIÓN Y SEGUIMIENTO VISUAL DEL TARGET	68
4.1 Detección, descripción y emparejamiento de puntos de interés del target.....	68
4.1.1 Evaluación de algoritmos de detección de puntos de interés del target	68
4.1.2 Discriminación del target en movimiento.....	71
4.2 Seguimiento visual y posición relativa del target	77
4.2.1 Umbral de decisión	77
4.2.2 Encapsulamiento del target.....	79

4.2.3 Problemas en la detección	80
4.3 Seguimiento del target de fotograma a fotograma.....	82
4.3.1 Proceso de detección y seguimiento	83
4.3.2 Refrescamiento de la posición del target	85
4.3.3 Posición virtual del target	88
CAPITULO V.....	90
SEGUIMIENTO Y ATERRIZAJE DEL VEHÍCULO AEREO NO TRIPULADO.....	90
5.1 Estimación del modelo de movimiento del drone	90
5.1.1 Consideraciones para el modelado	90
5.1.2 Modelamiento basado en estimación del movimiento entre imágenes	92
5.1.3 Obtención del modelo a partir de los datos acumulados entre imágenes	94
5.2 Diseño del controlador.....	99
5.3 Seguimiento y aterrizaje del UAV en la plataforma móvil terrestre	102
5.3.1 Implementación del controlador para seguimiento de la plataforma móvil terrestre	103
5.3.2 Seguimiento de la plataforma terrestre	108
5.3.3 Aterrizaje del UAV en la plataforma móvil terrestre	111
CAPITULO VI.....	118
PRUEBAS EXPERIMENTALES DE FUNCIONAMIENTO Y RESULTADOS.....	118
6.1 Pruebas experimentales dentro de los Hangares del CICTE	120
6.1.1 Desempeño del seguimiento visual	121
6.1.2 Desempeño del control de movimientos	123
6.2 Pruebas experimentales dentro del Coliseo Gral. Miguel Iturralde.....	128
6.2.1 Desempeño del seguimiento visual	128
6.2.2 Desempeño del control de movimientos	129
6.3 Pruebas experimentales en el patio exterior al Hangar del CICTE	133
6.3.1 Desempeño del seguimiento visual	133
6.3.2 Desempeño del control de movimientos	134

CAPÍTULO VII.....	138
CONCLUSIONES Y RECOMENDACIONES.....	138
7.1 Conclusiones.....	138
7.2 Recomendaciones	140
REFERENCIAS BIBLIOGRÁFICAS.....	142

ÍNDICE DE TABLAS

Tabla 1 <i>Secuencia de empuje para la configuración positiva (+).</i>	15
Tabla 2 <i>Secuencia de empuje para la configuración en cruz (x).</i>	16
Tabla 3 <i>Tópicos fundamentales publicados por el driver bebop_autonomy con sus descripciones.</i>	54
Tabla 4 <i>Parámetros de interés en la configuración del dron.</i>	56
Tabla 5 <i>Efecto de los parámetros del mensaje tipo geometry_msgs::Twist en el tópico cmd_vel.</i>	61
Tabla 6 <i>Efecto de los parámetros del mensaje tipo geometry_msgs::Twist en el tópico camera_control.</i>	61
Tabla 7 <i>Función de los métodos de la clase Bebop_mov.</i>	63
Tabla 8 <i>Parámetros y características para las subscripciones implementadas.</i>	66
Tabla 9 <i>Detección máxima de puntos de interés para los algoritmos SIFT, SURF y ORB.</i>	69
Tabla 10 <i>Medidas de eficiencia de los algoritmos SIFT, SURF y ORB en secuencias de video.</i>	75
Tabla 11 <i>Correspondencias con el target en espacio interior.</i>	77
Tabla 12 <i>Correspondencias con el target en espacio exterior.</i>	77
Tabla 13 <i>Acciones de control vinculadas al movimiento en cada eje del sistema de coordenadas.</i>	91
Tabla 14 <i>Consideraciones para la extracción de los parámetros de movimiento del dron.</i>	94
Tabla 15 <i>Características de interés en los escenarios de pruebas experimentales.</i>	119
Tabla 16 <i>Secuencia de acciones para cada escenario de pruebas experimentales.</i>	120
Tabla 17 <i>Evaluación de la detección y seguimiento visual para tres secuencias de aterrizaje a 0.5 m/s.</i>	121
Tabla 18 <i>Evaluación de la detección y seguimiento visual para tres secuencias de aterrizaje a 0.7 m/s.</i>	122
Tabla 19 <i>Error de posición (cm) al momento del aterrizaje para las secuencias de prueba en el hangar.</i>	126
Tabla 20 <i>Evaluación del seguimiento visual para cinco secuencias de aterrizaje a 0.8 m/s de la plataforma terrestre dentro del coliseo.</i>	129
Tabla 21 <i>Error de posición (cm) al momento del aterrizaje para las secuencias de prueba en el coliseo.</i>	132
Tabla 22 <i>Evaluación del seguimiento visual para tres secuencias de aterrizaje a 0.6 m/s de la plataforma terrestre en el patio exterior al Hangar del CICTE.</i>	134
Tabla 23 <i>Error de posición (cm) al momento del aterrizaje para las secuencias de prueba en el hangar.</i>	136

ÍNDICE DE FIGURAS

Figura 1. Estructura de descomposición del sistema (EDT) “VisualNavDrone”	2
Figura 2. Definición de los elementos del sistema.	11
Figura 3. Parámetros geométricos para multicopteros con diferentes configuraciones.	13
Figura 4. Sistema de coordenadas y modelo físico simplificado.	14
Figura 5. Giro alternado de las hélices para los motores de un Cuadricóptero.	14
Figura 6. Configuraciones para el movimiento del cuadricóptero.	15
Figura 7. Una imagen I representada como una superficie bidimensional.	17
Figura 8. Formación de la imagen con el modelo de Pin-Hole.	17
Figura 9. Modelo de Pin-hole.	18
Figura 10. Proyección de un Punto P de la escena 3D en un plano de imagen 2D pasando a través del centro óptico y una distancia focal.	19
Figura 11. Modelo de cámara. Proyección de puntos en la imagen.	20
Figura 12. Representación Global y Local de las características de una imagen.	22
Figura 13. Ejemplo puntos de interés de una imagen.	23
Figura 14. Clasificación de los puntos en la imagen basada en los valores propios de la matriz de autocorrelación M.	27
Figura 15. Búsqueda de los extremos en 3D de espacio escala con DOG.	27
Figura 16. Comparación de intensidad de un pixel mediante FAST en un círculo de radio 3.	29
Figura 17. Comparación del punto con sus 26 vecinos más cercanos.	33
Figura 18. Ejemplo de descriptor del punto de interés en una ventana de 8×8 que genera un descriptor de 2×2	34
Figura 19. En la mitad izquierda derivadas parciales Gaussianas de segundo orden en dirección y, xy. En la mitad derecha el uso de filtros de caja; la región negra es 0.	35
Figura 20. Filtros de Haar para la respuesta en x, y. La parte negra tiene peso -1 y la blanca +1.	36
Figura 21. Asignación de la orientación en ventanas de $\pi/3$	36
Figura 22. Ilustración a modo de ejemplo del descriptor de SURF con regiones de 4×4 y subregiones de 2×2 y las respuestas al filtro Haar dx, dy.	37
Figura 23. Distorsiones que surgen de la transformación afín plana. a) Rotación por $R(\theta)$. b) Deformación con el ángulo y rotación asociados $R(\varphi)$	41
Figura 24. Distorsiones de proyección central debido a (a) Transformación de Similitud. (b) Transformación Afín. (c) Transformación de Perspectiva.	42
Figura 25. Esquemas de control. (a) En lazo abierto (b) En lazo cerrado.	43
Figura 26. Diagrama de bloques del controlador PID en modo paralelo.	44
Figura 27. Bebop 1 de Parrot con las protecciones de espacio interior EPP («Parrot gallery», s. f.).	46

Figura 28. Dimensiones del drone Bebop 1 de Parrot en una vista cenital y lateral.	47
Figura 29. Movimientos del drone Bebop 1 de Parrot para cada acción de control.	47
Figura 30. Ángulos de apertura de video para la cámara del Bebop 1 desde la perspectiva cenital y lateral.	48
Figura 31. Dimensiones de la plataforma móvil terrestre.	49
Figura 32. Imagen consigna elegida para el seguimiento (target).	50
Figura 33. Montaje final de la plataforma móvil terrestre.	50
Figura 34. Relación de aspecto entre la plataforma de aterrizaje y el drone donde se aprecia la permisibilidad de posicionamiento.	51
Figura 35. Íconos correspondientes de Ubuntu y OpenCV.	52
Figura 36. Ícono del sistema operativo de Robots (ROS).	52
Figura 37. Esquema del funcionamiento básico de ROS con un nodo talker y uno listener comunicados gracias a un master (roscore).	53
Figura 38. Gráfico dinámico del proceso de subscripción a imágenes en rqt_graph.	58
Figura 39. Diagrama de conversión de imágenes entre ROS y OpenCV.	58
Figura 40. Diagrama UML de la clase Bebop_mov.	62
Figura 41. Localización del sensor de flujo óptico para estimación de la velocidad del drone.	64
Figura 42. Relación detección-tiempo de los algoritmos SIFT, SURF y ORB para la secuencia de imágenes de la Tabla 9.	70
Figura 43. Detección de 25 puntos de interés en la imagen del target. (a) Algoritmo SIFT. (b) Algoritmo SURF. (c) Algoritmo ORB.	70
Figura 44. Diagrama de bloques de la detección del target en una secuencia de fotogramas.	73
Figura 45. Medidas de recall para cada fotograma consecutivo en la secuencia de video 2 con los algoritmos SIFT, SURF y ORB.	75
Figura 46. Medidas de precisión para cada fotograma consecutivo en la secuencia de video 2 con los algoritmos SIFT, SURF y ORB.	76
Figura 47. Medidas de tiempo de procesamiento para cada fotograma consecutivo en la secuencia de video 2 con SIFT, SURF y ORB.	76
Figura 48. Correspondencias obtenidas en espacio interior con respecto a la referencia de 70 puntos.	78
Figura 49. Correspondencias obtenidas en espacio exterior con respecto a la referencia de 70 puntos.	78
Figura 50. Encapsulamiento del target con la transformación de perspectiva.	80
Figura 51. Errores en la detección por: (a) escala, (b) baja calidad en imagen y (c) cambios de textura e iluminación.	81
Figura 52. Diagrama de Flujo de la propuesta de detección y seguimiento visual de fotograma a fotograma.	82
Figura 53. Correspondencia de puntos de interés entre fotogramas consecutivos.	84
Figura 54. Secuencia de imágenes con deformaciones del recuadro.	85

Figura 55. Secuencia de imágenes con refrescamiento del target y corrección del recuadro. (Izquierda) fotograma anterior al refrescamiento, (centro) porción de imagen que contiene al target en la escena, (derecha) fotograma siguiente con corrección del recuadro.	86
Figura 56. Diagrama de flujo del seguimiento del target entre fotogramas consecutivos con refrescamiento periódico de la posición.	87
Figura 57. Target virtual representado por transformaciones entre fotogramas consecutivos.	88
Figura 58. Movimientos respecto al target representados como un error de posición.	89
Figura 59. Bloques de la obtención de parámetros de movimiento del dron.	92
Figura 60. Transformaciones geométricas entre imágenes consecutivas.	93
Figura 61. Levantamiento del plano de imagen con respecto al plano de tierra $\beta = 20^\circ$	94
Figura 62. Comportamiento del dron con órdenes de control en Roll. Superior: entrada. Inferior: salida.	96
Figura 63. Comportamiento del dron con órdenes de control en Pitch. Superior: entrada. Inferior: salida.	96
Figura 64. Identificación de la Planta en Roll.	98
Figura 65. Identificación de la Planta en Pitch.	98
Figura 66. Esquema de control para el seguimiento del target mediante la realimentación de la posición visual.	99
Figura 67. Esquema de diseño del controlador.	99
Figura 68. Respuesta al impulso unitario del sistema controlador planta en Roll.	101
Figura 69. Respuesta al impulso unitario del sistema controlador planta en Pitch.	102
Figura 70. Diagrama de flujo del aterrizaje automático en superficies móviles.	103
Figura 71. Error de posición del centro de masa del target con respecto al centro de la imagen.	105
Figura 72. Longitud de detección del eje de las ordenadas sobre el plano de tierra a la altura de un metro.	107
Figura 73. Longitud de detección del eje de las abscisas sobre el plano de tierra a la altura de un metro.	107
Figura 74. Secuencia de imágenes de la detección y seguimiento de la plataforma terrestre.	109
Figura 75. Posición del dron en cada instante de tiempo desde la detección inicial hasta la final.	110
Figura 76. Trayectoria recorrida por el dron en el proceso de seguimiento.	110
Figura 77. Fijación del nuevo punto como centro para seguimiento en la parte superior del recuadro.	112
Figura 78. Umbral de aterrizaje definido (las medidas se encuentran en cm).	113
Figura 79. Proyección del centro óptico de la cámara del dron a una altura de 40 cm y 70° de descenso de la cámara virtual.	114

Figura 80. Zona de posicionamiento del drone debido al umbral definido. (a) Zona del centro del drone. (b) Zona máxima ocupada por todas las dimensiones del drone.	115
Figura 81. Secuencia de imágenes desde la detección del target hasta el aterrizaje dentro de la plataforma.	116
Figura 82. Posición del drone en cada instante de tiempo desde el despegue hasta el aterrizaje.	116
Figura 83. Trayectoria recorrida por el drone en el proceso de aterrizaje.	117
Figura 84. Secuencia de imágenes correspondientes a las pruebas experimentales en los Hangares del CICTE.	120
Figura 85. Medición del desplazamiento generado con respecto al deseado en el tiempo en el proceso de aterrizaje para la secuencia 6.	123
Figura 86. Error de desplazamiento en el control del drone para la secuencia 6. ...	124
Figura 87. Error de posición en metros desde el drone hasta el target en los ejes X, Y para la secuencia 6.	125
Figura 88. Error de desplazamiento en cada instante de tiempo para las secuencias de prueba en el hangar.	125
Figura 89. Error de posición en metros desde el drone hasta el target en los ejes X, Y para las secuencias de prueba en el hangar.	126
Figura 90. Imágenes tomadas antes del aterrizaje para las secuencias de prueba en el hangar.	127
Figura 91. Secuencia de imágenes correspondientes a las pruebas experimentales en el Coliseo Gral. Miguel Iturralde.	128
Figura 92. Medición del desplazamiento generado con respecto al deseado en el tiempo en el proceso de aterrizaje para la secuencia 2.	130
Figura 93. Error de desplazamiento en cada instante de tiempo para las secuencias de prueba en el coliseo.	130
Figura 94. Error de posición en metros desde el drone hasta el target en los ejes X, Y para las secuencias de prueba en el coliseo.	131
Figura 95. Imágenes tomadas antes del aterrizaje para las secuencias de prueba en el coliseo.	132
Figura 96. Secuencia de imágenes correspondientes a las pruebas experimentales en el patio exterior a los hangares del CICTE.	133
Figura 97. Medición del desplazamiento generado con respecto al deseado en el tiempo en el proceso de aterrizaje para la secuencia 1.	135
Figura 98. Error de desplazamiento en cada instante de tiempo para las tres secuencias exteriores.	135
Figura 99. Error de posición en metros desde el drone hasta el target en los ejes X, Y para las secuencias de prueba exterior.	136
Figura 100. Imágenes tomadas antes del aterrizaje para las secuencias de prueba en el patio fuera del coliseo.	137

RESUMEN

El presente trabajo de investigación aborda un sistema para el aterrizaje automático de un vehículo aéreo no tripulado (UAV) en una plataforma terrestre en movimiento, los algoritmos de control son implementados en una estación en tierra que se comunica de manera remota con el UAV, el sistema tiene como principal recurso la información visual proporcionada por una cámara a bordo del UAV. Las imágenes obtenidas son procesadas mediante la caracterización de imágenes donde se realiza la detección, descripción y emparejamiento de puntos de interés con el fin de discriminar la plataforma móvil y realizar el seguimiento de la misma. En el desarrollo del proyecto se realiza la comparación de tres de los algoritmos más conocidos en la caracterización de imágenes SIFT, SURF y ORB para determinar el de mejor desempeño en la aplicación, posteriormente se describe la propuesta de seguimiento visual de un objetivo mediante las transformaciones geométricas aplicadas a la caracterización de fotogramas consecutivos con un ciclo de refrescamiento, logrando mayor exactitud en la localización del objetivo y disminución de las incertidumbres en la detección por cambios de visualización inter-fotograma y condiciones ambientales. Se realiza la estimación del modelo dinámico de movimiento del drone con base en la estimación de los parámetros de movimiento inter-fotograma obtenidos en post procesamiento para presentar finalmente la propuesta de seguimiento y aterrizaje automático en la plataforma móvil. El sistema ha sido sometido a pruebas experimentales en tres escenarios de diferentes características presentando un correcto desempeño a pesar de las perturbaciones del ambiente.

PALABRAS CLAVE:

- **CARACTERIZACIÓN**
- **TRANSFORMACIÓN GEOMÉTRICA**
- **ESTIMACIÓN DE MOVIMIENTO**
- **SISTEMA DE CONTROL**
- **ATERRIZAJE AUTOMÁTICO**

ABSTRACT

This research thesis is focused on development of an automatic landing system for unmanned aerial vehicles (UAV) on a ground motion platform, the control algorithms are implemented in a ground station communicated with the UAV. The main input data of the system is visual information from the on-board camera of the UAV. The acquired frames are processed by image characterization where the feature points detection, description and matching is necessary for mobile platform tracking. Three widely used algorithms for detection and description (SIFT, SURF and ORB) were compared in order to determine highest performance for the application. The target tracking proposal is based on geometric transforms between frame and frame with a refresh cycle, increasing the accuracy in target localization and decreasing detection uncertainties due to inter-frame warping and environment conditions. UAV motion dynamic model estimation is obtained by frame to frame motion estimation parameters, and used for automatic tracking and landing control system on the mobile platform. The system has tested in three different scenarios, and shows a high performance despite environment disturbances.

KEYWORDS:

- **IMAGE FEATURE**
- **GEOMETRIC TRANSFORM**
- **MOTION ESTIMATION**
- **CONTROL SYSTEM**
- **AUTOMATIC LANDING**

CAPITULO I

GENERALIDADES

1.1 Planteamiento del Problema

La investigación forma parte del proyecto titulado “Sistemas de percepción y localización para la navegación autónoma de vehículos aéreos multirrotores de micro escala en entornos GPS-denegados (VisualNavDrone)” perteneciente al CICTE (Centro De Investigación Científica y Tecnológica Del Ejército) y dirigido por el Dr. Wilbert G. Aguilar. El objetivo del proyecto es desarrollar sistemas de navegación basados en la combinación de datos inerciales y movimiento estimado con algoritmos de visión por computadora. Estos sistemas serán utilizados para la navegación autónoma de drones en escenarios en los que se pierda la comunicación y la navegación semiautónoma para entornos GPS-denegados.

El proyecto propone la combinación de datos provenientes de los sensores inerciales con el movimiento estimado a partir de imágenes capturadas con cámaras a bordo del dron para la navegación autónoma en casos en que se requiera, y dando soporte a la navegación semiautónoma para minimizar el tiempo de entrenamiento de los tele-operadores y maximizar su desempeño durante el pilotaje, además de incluir: localización y mapeado visual, detección y seguimiento de objetivos, evasión de obstáculos, aterrizaje automático, y en el caso de pérdida de comunicación retorno autónomo al punto de partida o al punto de pérdida.

En la Figura 1 se muestra una estructura de descomposición de trabajo (EDT) en la que se detallan las etapas del proyecto de investigación.

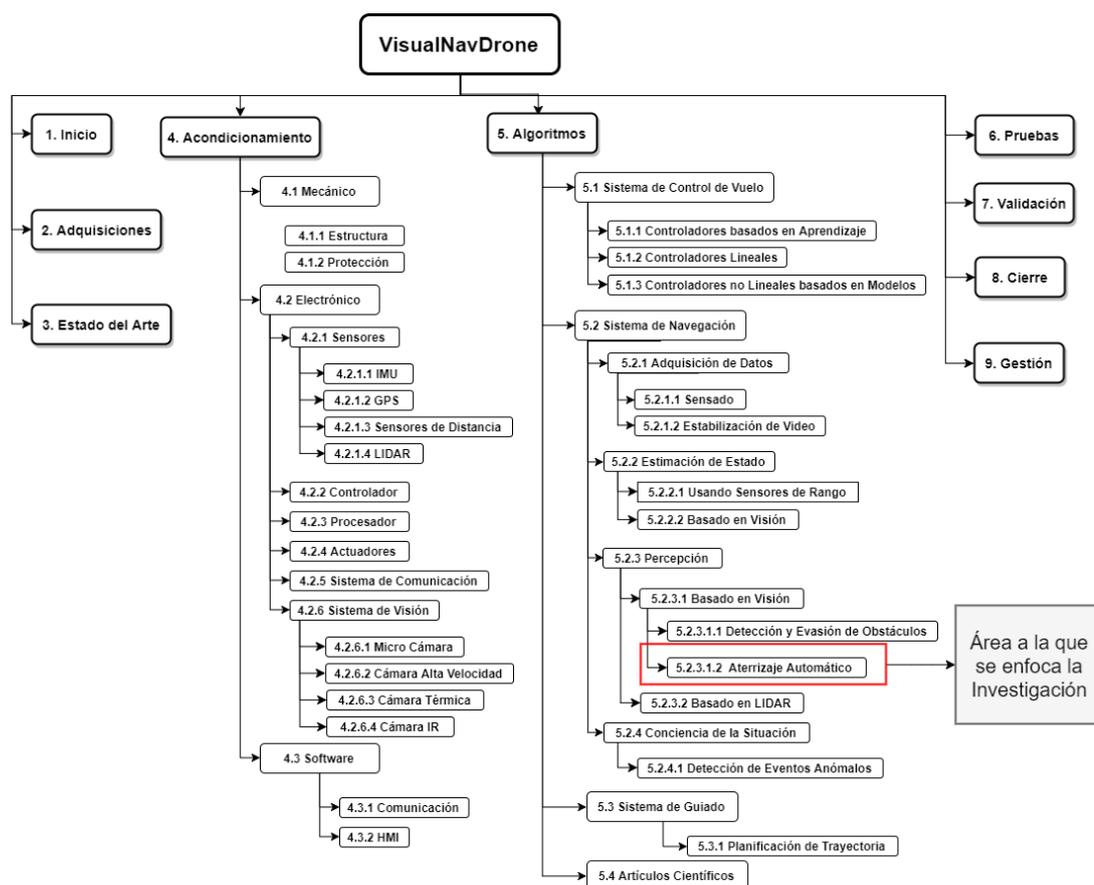


Figura 1. Estructura de descomposición del sistema (EDT) “VisualNavDrone”

La etapa 5 (Figura 1) corresponde al desarrollo de los algoritmos de control de vuelo, navegación y guiado, que son parte fundamental dentro de los sistemas de navegación autónoma (VisualNavDrone) dando cabida a que el Sistema incluya:

- Localización y mapeado visual
- Detección y seguimiento de objetivos
- Evasión de obstáculos
- Aterrizaje automático
- Retorno autónomo al punto de partida o al punto de pérdida (para cuando se dé el caso de pérdida de comunicación)

El presente proyecto de investigación se enfocará en el Aterrizaje automático de UAVs como complemento al proyecto “VisualNavDrone” para vehículos aéreos multirotor en caso que tengan la necesidad de realizar el aterrizaje en plataformas bajo condiciones de movimiento.

1.2 Antecedentes

El uso de vehículos aéreos no tripulados UAV (Unmanned Aerial Vehicle por sus siglas en inglés) ha sido significativo a lo largo de las últimas décadas debido a las múltiples aplicaciones tanto de investigación como militares que justifican la demanda de los mismos (Kendoul, 2012; Limnaios, 2014). En (Kendoul, 2012) se realiza una diferenciación en cinco clases de UAVs según su tamaño y su carga útil de las cuales es de interés en esta investigación la categoría V de micro UAV.

Hoy los UAV son comunes en el aire abordando aplicaciones militares como en servicio público, vigilancia, monitoreo y otros (Aguilar, Luna, Moya, et al., 2017, 2017, 2017; Aguilar & Angulo, 2014; Limnaios, 2014); donde los micro UAV toman la delantera por poder volar en entornos cerrados y pequeños con múltiples sensores y cámaras que permiten sistemas de control que fortalecen su navegación, como el caso de los sistemas basados en las imágenes de los dispositivos de visión a bordo para mejorar la autonomía y robustez. Además sus características han generado una gran ventaja en aplicaciones civiles como la detección de eventos, reconocimiento de objetos, suelos de cultivo, tráfico vehicular, modelamiento 3D y otros (Limnaios, 2014) por su versatilidad y el reemplazo del enfoque tradicional de un “tripulante que controla al vehículo” por un tele-operador, disminuyendo los riesgos en el vuelo.

El control remoto efectivo de un drone está sujeto a varios criterios entre los cuales se incluye la suficiente adquisición de datos y tiempo para la toma de decisiones (Clarke, 2014) y aunque hasta la década de los 90 la principal aplicación de estos vehículos era militar con la inspección, mapeo y exploración de áreas hostiles (Achille et al., 2015) (Chmaj & Selvaraj, 2015), el crecimiento tecnológico de los UAV ha ido generando nuevos campos de investigación e innovadores algoritmos de navegación con un solo objetivo: lograr la navegación autónoma de los UAV's y ampliar el alcance de los mismos para una proyección a aplicaciones de peligro. Además, el mal manejo de las naves puede causar accidentes, por lo cual ya sean UAV's de escala micro o macro, hay la necesidad de desarrollar algoritmos que apoyen la navegación y aterrizaje de estos.

Algunos de los principales recursos de adquisición de información que tienen a bordo los UAV son la cámara a bordo, la IMU (Unidad de Medición Inercial) y el GPS (Sistema de Posicionamiento Global), a pesar de ello hay ambientes en los cuales no se puede contar con el GPS o hay problemas en la IMU (Duffy et al., 2017) quedando únicamente las imágenes como retroalimentación de la posición y navegación. Para estos casos se necesita la captura correcta de imágenes (Whitten, 2017) y el procesamiento robusto y confiable con una disminución de efectos indeseados en las imágenes para asegurar la seguridad durante el vuelo (Aguilar & Angulo, 2014, 2014, 2016).

Sistemas innovadores para su control han sido introducidos gracias al diseño de estos vehículos, sus medidas y capacidades como se muestra en (Hassanalian & Abdelkefi, 2017) el control mediante google glass, brain computer interface o mandos a distancia de gran alcance, pero siguen siendo susceptibles a los cambios climáticos que delimitan su uso (Custers, 2016) (Charrette, Royer, & Chausse, 2016).

El aterrizaje de vehículos aéreos no tripulados de manera automática ha sido un tema poco abordado por la comunidad científica, algunos trabajos como (Azinheira & Rives, 2008) buscan el reconocimiento y aterrizaje automático basado en modelado de imágenes y estimación de movimiento y posición para compensar el desplazamiento requerido obteniendo como resultado un error apreciable en el despegue y con 0.2° y 1.2° de error en giro al aterrizaje final. En (Wenzel, Masselli, & Zell, 2011) realizan el seguimiento y aterrizaje en una superficie en movimiento con el uso de una marca artificial sobre un carrito pequeño que se desplaza circularmente. Utilizan una cámara remota infrarroja para estabilizar el control de vuelo del UAV sobre el vehículo, todo el procesamiento es realizado sobre un microcontrolador Atmel ATmega 644P.

Como se puede apreciar en (Singh & Padhi, 2009) un correcto aterrizaje está basado principalmente en la elección de una trayectoria adecuada de manera que en ocasiones no basta con un modelo lineal, en éste artículo se opta por un modelo de inversión dinámica para el aterrizaje automático para lograr controlar la altura de aterrizaje, la dirección y la disminución de altura para estacionar.

Algunas de las herramientas fundamentales para realizar control sobre la trayectoria, y seguimiento a ciertos objetivos son los algoritmos de detección de puntos de interés, tales como SIFT, PCA-SIFT, SURF, KAZE, BRISH y ORB; en (Juan & Gwun, 2009) se puede ver una comparación de los tres primeros algoritmos. Con el uso de alguna de las herramientas mencionadas anteriormente y mediante el procesamiento continuo de imágenes, se puede realizar un estimativo del movimiento realizado.

Múltiples propuesta para navegación autónoma de UAVs se basan en dos componentes principales:

- Percepción: Sistema visual SLAM con cámaras monoculares (Engel, Sturm, & Cremers, 2012a, 2012b, 2014; Urzua, Munguía, & Grau, 2017) o cámaras RGB-D (Aguilar, Rodríguez, & Álvarez, 2017; Aguilar, Rodríguez, Álvarez, Sandoval, & Quisaguano, 2017, 2017) para mapeo en tiempo real, y un filtro extendido de Kalman para la fusión de datos del sensor y estimación de estado del UAV.
- Seguimiento de Trayectoria: Un planificador de trayectorias basados en mapas probabilísticos o árboles de búsqueda (Aguilar & Morales, 2016; Aguilar, Morales, Ruiz, & Abad, 2017, 2017) y un control PID para el seguimiento de trayectoria.

Cabe mencionar que el mapeo visual al utilizar algoritmos como SLAM conlleva un mayor costo computacional haciendo del proceso más exigente para el equipo.

Varios controladores pueden ser utilizados para control de trayectoria entre los que se puede mencionar al PID, no obstante trabajos como (Nho & Agarwal, 2000) muestran aplicación de control difuso para el control del aterrizaje automático en modelos lineales y no lineales; combinado con control en modo de deslizamiento para modificar los factores de escala haciendo que tome la trayectoria de vuelo correcta.

Tomando en cuenta estos antecedentes, y con el objetivo de superar las dificultades expuestas y desafíos en el problema de aterrizaje para superficies móviles, se propone en esta investigación un sistema de aterrizaje automático utilizando procesamiento de imágenes para reconocimiento y estimación de la posición del vehículo móvil terrestre,

estimación del movimiento generado por detección de puntos de interés, seguimiento de trayectorias y compensación de movimiento por posibles perturbaciones externas como el viento para el aterrizaje seguro.

1.3 Justificación e importancia

En la última década la comercialización de vehículos aéreos no tripulados para aplicaciones civiles se ha incrementado notablemente. No obstante debido a su bajo costo han sido adquiridos por personas sin la adecuada formación para tele-operar este tipo de plataformas. Esto incrementa considerablemente el riesgo de accidentes, por lo que distintos grupos de investigación se encuentran desarrollando algoritmos de navegación autónoma que mitiguen ese riesgo (Kendoul, 2012). Los sistemas de navegación autónoma no solo dan soporte a tele-operadores sin experiencia sino que pueden ser aplicados en entornos de difícil acceso o con condiciones de alto riesgo (Maza, Caballero, Capitán, Martínez-De-Dios, & Ollero, 2011; Papakonstantinou, Topouzelis, & Pavlogeorgatos, 2016) donde se realiza el reconocimiento aéreo de desastres o para la seguridad civil.

Uno de los principales problemas a ser solventados dentro de los sistemas de navegación autónoma es el aterrizaje automático en superficies móviles, en caso de entornos GPS-denegados, fallas de comunicación con el UAV o fallas en la tele-operación del equipo. En el desarrollo de este proyecto de investigación se va a realizar un sistema de aterrizaje automático basado en visión artificial y sensores inerciales enfocado a resolver este problema, contribuyendo al proyecto “Sistemas de percepción y localización para la navegación autónoma de vehículos aéreos multirrotores de micro escala en entornos GPS-denegados (VisualNavDrone)” realizado por el Centro de Investigación Científica y Tecnológica del Ejército (CICTE), contribución que ayudará al aterrizaje seguro en plataformas con distintas condiciones de movimiento que se ajusten a los alcances del proyecto.

El aterrizaje automático es parte de múltiples aplicaciones de carácter militar, científico o civil como: aterrizaje automático en superficies de cuerpos celestes (Kendoul, 2012, p. 337), condiciones de mal tiempo (vientos de alta velocidad o movimiento de la plataforma de aterrizaje) o en caso de una avería en vehículos

tripulados para evitar que se estrelle. En la literatura se pueden encontrar múltiples investigaciones basadas en aterrizaje automático de UAVs sobre distinto tipo de superficies, no obstante, la mayoría no toma en cuenta condiciones en movimiento de la plataforma de aterrizaje, salvo ciertos trabajos como (Wenzel et al., 2011) donde se usan sensores IR para el seguimiento de la misma o el proyecto “EC-SAFEMOBIL” donde se aterriza un UAV sobre una plataforma que se mueve en el mismo sentido y de manera rectilínea al movimiento del mismo.

Por esta razón, se pretende realizar un sistema que abarque el aterrizaje automático sobre superficies en movimiento mediante visión artificial, dando cabida a múltiples aplicaciones potenciales, brindando seguridad y evitando accidentes o pérdidas del equipo por errores en la operación del vehículo o aterrizajes inadecuados.

1.4 Alcance del proyecto

El presente trabajo de investigación tiene como objetivo un sistema del aterrizaje automático de un vehículo aéreo no tripulado multirrotor de la categoría V (definida en (Kendoul, 2012)), en plataformas en movimiento. Por cuestiones de seguridad este trabajo se enfocará en movimientos de la plataforma de aterrizaje no superiores a 1 m/s, la trayectoria de la plataforma móvil estará definida por movimientos rectilíneos entre puntos no definidos. Pese a perturbaciones externas, el vehículo podrá recuperar la trayectoria original de aterrizaje mediante compensación del movimiento. Las perturbaciones a las que se someta el UAV no deben alcanzar niveles de riesgo para la integridad del vehículo.

El control de un UAV multirrotor se lo realizará con base en las imágenes capturadas por la cámara monocular y la información inercial de la IMU. A partir de las imágenes obtenidas se estimarán los parámetros de movimiento que servirán de consigna para el control de trayectoria.

Como parte del procesamiento de imágenes se utilizarán algoritmos de detección y descripción de puntos de interés, se estimarán transformaciones geométricas entre fotogramas consecutivos y posteriormente se desestimarán las correspondencias que no se acoplen al modelo de movimiento entre fotogramas.

La estimación de movimiento del UAV será establecida mediante los parámetros de movimiento que se obtienen del procesamiento de imágenes, no obstante, es necesario el uso de un sistema de control para la compensación de movimientos y seguimiento de trayectorias. Los algoritmos desarrollados serán implementados en una estación en tierra con Sistema Operativo ROS, la cual se comunicará con el vehículo aéreo no tripulado inalámbricamente por wifi. Finalmente se proceden a realizar pruebas de funcionamiento que permitan determinar el desempeño del sistema.

1.5 Objetivos

1.5.1 Objetivo General

Desarrollar un sistema de control basado en visión artificial para el aterrizaje automático de un vehículo aéreo no tripulado sobre una superficie móvil mediante el seguimiento de puntos de interés.

1.5.2 Objetivos Específicos

- Estimar los parámetros de movimiento entre fotogramas consecutivos como entrada del sistema de control del seguimiento de trayectorias durante el aterrizaje automático basado en las correspondencias entre imágenes consecutivas.
- Identificar el modelo de movimiento del UAV para trayectorias rectilíneas de la plataforma móvil basado en los parámetros de movimiento estimados entre fotogramas consecutivos
- Diseñar un sistema de control que permita compensar los movimientos requeridos para el seguimiento del objetivo o debidos a perturbaciones externas.
- Implementar los algoritmos desarrollados en una estación en tierra que se comunique y controle al UAV.
- Analizar los resultados obtenidos en pruebas experimentales.

1.6 Descripción del proyecto de investigación

El proyecto de investigación consiste en la elaboración de un algoritmo que basado en la detección de puntos de interés en las imágenes tomadas desde un vehículo aéreo no tripulado (en este caso el dron Bebop 1 de Parrot) pueda realizar el reconocimiento de la plataforma de aterrizaje y el seguimiento de la misma para poder aterrizar en ella de manera efectiva.

La plataforma móvil terrestre realiza movimientos rectilíneos entre puntos no definidos y por seguridad alcanza velocidades no superiores a 1 m/s de manera que el seguimiento y aterrizaje se realizan a esa velocidad como máximo. El vehículo aéreo no tripulado ha sido configurado de manera que su cámara virtual apunte hacia el piso teniendo un ángulo de descenso de 70° , es decir el plano de la imagen tendría 20° de elevación con respecto al plano de tierra (el piso).

Para acceder a la información proporcionada por el dron se ha recurrido a la utilización del controlador oficial de ROS para el dron Bebop 1 y 2 de Parrot "Bebop_autonomy" el cual traduce las múltiples funcionalidades del dron en nodos, tópicos, servicios y parámetros. Se han implementado un conjunto de funciones que facilitan la comunicación con el controlador y permiten acceder a ellas desde el programa principal.

Para reconocimiento y seguimiento visual de la plataforma de aterrizaje se ha realizado una comparación de tres de los algoritmos más representativos en la caracterización de imágenes SIFT, SURF y ORB para determinar la mejor opción dada la aplicación según su desempeño. Se analiza también el umbral de decisión que permite determinar si el objetivo ha sido reconocido satisfactoriamente.

La posición del objetivo es estimada mediante una transformación geométrica de perspectiva a las correspondencias que han sido satisfactoriamente emparejadas en el proceso de caracterización de imágenes, representándola visualmente como un recuadro que envuelve al objetivo.

Para la correcta detección y seguimiento visual del objetivo se analiza la caracterización de secuencias de fotogramas con una imagen como consigna y con la propuesta de caracterización tomando el fotograma anterior como consigna, dando una mejor detección y posicionamiento del objetivo con el segundo método debido a que realiza la caracterización entre imágenes con el menor cambio permitido por la capacidad de captura de la cámara. Para concluir con la detección y seguimiento visual se realiza un refrescamiento periódico de la posición del objetivo, eliminando así el error acumulado en la posición.

Debido a las limitaciones del UAV se detalla e implementa la propuesta de modelamiento del dron con base en los datos provenientes de la estimación de los parámetros de movimiento inter-fotograma obtenidos del procesamiento de imágenes consecutivas para encontrar correspondencias y posteriormente estimar transformaciones geométricas que determinen dichos parámetros como traslaciones en el espacio.

Para seguir la trayectoria del objetivo en movimiento se realiza el diseño e implementación de un controlador PID para el modelo del dron obtenido, donde se pretende el seguimiento de un delta de desplazamiento proporcional al error de posición del dron con respecto a la plataforma móvil en los ejes x , y , es decir el plano paralelo al piso, dejando de lado el control de altura durante el seguimiento.

Una vez comprobado el seguimiento satisfactorio durante un tiempo determinado se realiza la secuencia de aterrizaje con un descenso a velocidad constante donde el error de posición es estimado tomando en cuenta la altura de detección. El controlador PID se utiliza en la compensación del movimiento requerido para posicionar al dron en posición de aterrizaje final, el umbral para dicha posición está definido por un análisis de las dimensiones del dron y su posicionamiento en la plataforma.

En la Figura 2 se muestra un esquema de ejemplo del sistema donde el UAV persigue al objetivo visual y es controlado desde una estación en tierra mediante comunicación inalámbrica.

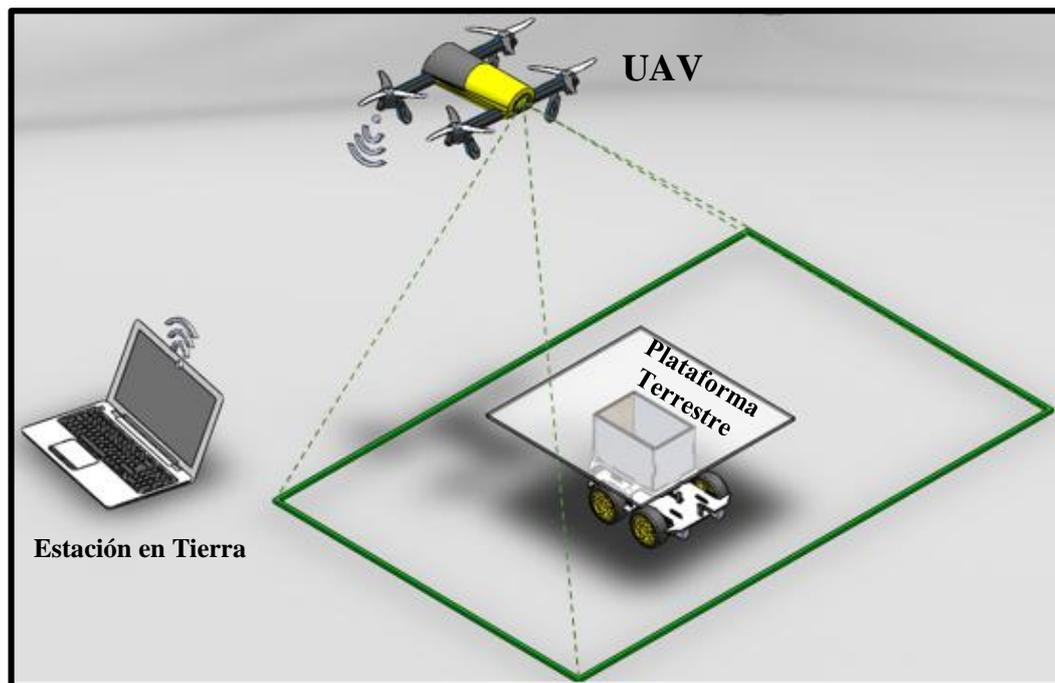


Figura 2. Definición de los elementos del sistema.

CAPITULO II

MARCO TEÓRICO

2.1 Conceptos generales de los Vehículos Aéreos Multirotores no Tripulados

2.1.1 Parámetros físicos

Los UAV (Unmanned Aerial Vehicle, por sus siglas en inglés) son vehículos aéreos que pueden solventar su estabilidad y vuelo sin necesidad de tripulantes en el mismo, pueden ser controlados de manera remota desde una estación en tierra, desde otro vehículo e incluso ejecutar maniobras de manera autónoma con la electrónica instalada en su estructura (Quan, 2017). Los UAV's constan a breves rasgos de un cuerpo, donde se ubican la batería y electrónica del mismo y de "n" conjuntos motor-hélice separados de manera equidistante del cuerpo.

Los brazos de un vehículo aéreo multirotor son los que sostienen a cada conjunto motor-hélice. El ángulo entre los brazos de un vehículo aéreo multirotor depende de cuantos conjuntos motor hélice deben sostenerse, para el caso del cuadricóptero la separación angular entre los brazos es de 90° cumpliendo con la ecuación (1) (Quan, 2017).

$$\theta = \frac{360}{n} \quad (1)$$

Donde "n" es la cantidad de conjuntos motor-hélice, además el radio de la estructura R está relacionado con el radio máximo de una hélice $r_{m\acute{a}x}$ como se aprecia en (2):

$$R = \frac{r_{m\acute{a}x}}{\sin\frac{\theta}{2}} = \frac{r_{m\acute{a}x}}{\sin\frac{180}{n}} \quad (2)$$

Las configuraciones de UAV's con respecto a la cantidad de conjuntos motor-hélice se pueden apreciar en la Figura 3.

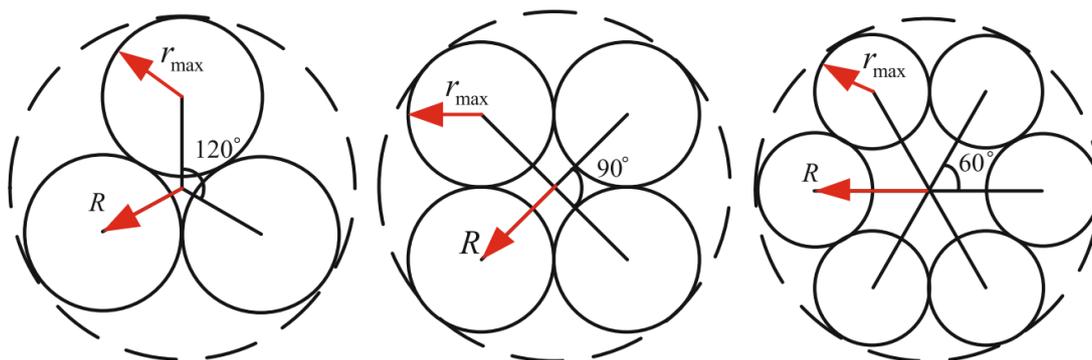


Figura 3. Parámetros geométricos para multirótores con diferentes configuraciones.

Fuente: (Quan, 2017)

El presente trabajo se ve enfocado a un Cuadrirótero, un tipo de UAV con cuatro conjuntos motor-hélice separados de manera equidistante del cuerpo, sujetos al mismo en forma de cruz como lo especifica (Leos Monroy, 2014).

2.1.2 Dinámica del movimiento

La dinámica del cuadrirótero se logra mediante la intervención de fuerzas y momentos que permiten los movimientos en el espacio.

La dinámica del movimiento del sistema es directamente influenciada por el giro de los motores a una velocidad w_i . La acción de las hélices sobre el aire generan una fuerza de empuje f_i , que se evidencia sobre cada punto central de la hélice con un momento relacionado T_{Mi} (Berrutti, Falkenstein, & Favaro, 2015). Por esta razón la generación del movimiento en el sistema depende de la velocidad de giro de los rotores.

La Figura 4 permite apreciar la acción de las fuerzas y momentos en el movimiento del sistema con respecto a los ejes de rotación en pitch, yaw y roll.

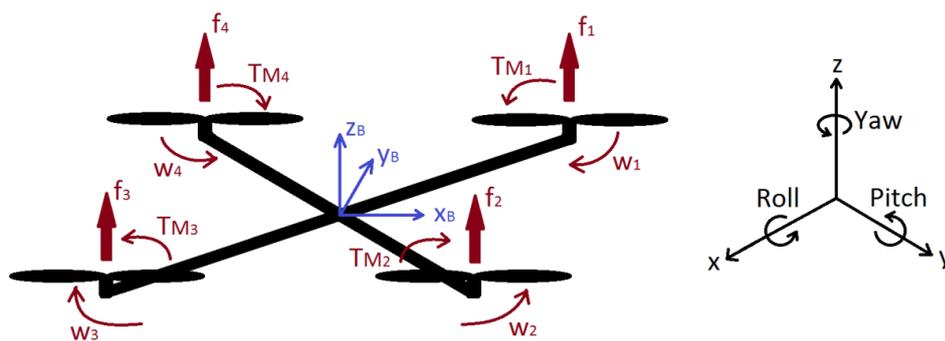


Figura 4. Sistema de coordenadas y modelo físico simplificado.

Fuente: (Berrutti et al., 2015)

El sentido de giro de las hélices en el cuadricóptero debe ser de manera alternada entre ellos, es decir, dos hélices de posición opuesta deben girar en sentido horario y las otras dos en sentido anti horario como se observa en la Figura 5, de manera que los momentos de los rotores sean cancelados en las proporciones que se desee para generar maniobras como: vuelo estático, axial, auto rotación y de avance.

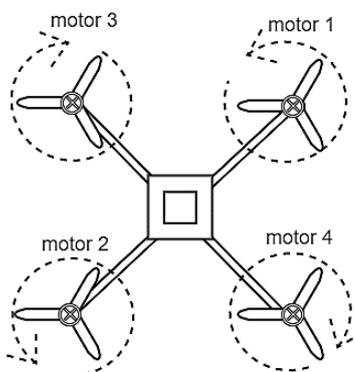


Figura 5. Giro alternado de las hélices para los motores de un Cuadricóptero.

Fuente: (Leos Monroy, 2014)

Los movimientos dependen de la secuencia de empuje generada por la diferencia de velocidad angular entre los rotores y se define según la configuración del cuadricóptero. Hay dos tipos de configuraciones para el movimiento de un cuadricóptero, la una es en cruz (x) y la otra es definida como positiva (+) (Figura 6) (Quan, 2017), estas configuraciones permiten gran maniobrabilidad de la plataforma.

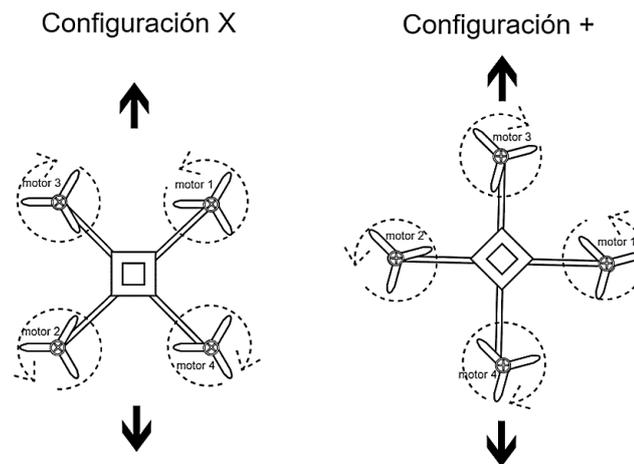


Figura 6. Configuraciones para el movimiento del cuadricóptero

Fuente: (Leos Monroy, 2014)

Las secuencias de empuje se ven reflejadas en el desbalance de la velocidad de giro entre los cuatro rotores, de manera que para generar el ascenso, descenso y los movimientos en los ejes de rotación Yaw, Pitch, Roll se deben generar las secuencias correspondientes a la Tabla 1 y Tabla 2 para cada configuración de movimiento en las cuales se usan w_i como la velocidad angular del respectivo rotor y Δ como la variación en igual proporción para los diferentes rotores.

Tabla 1

Secuencia de empuje para la configuración positiva (+).

Movimiento	Rotor 1	Rotor 2	Rotor 3	Rotor 4
Ascenso	$w_1 + \Delta$	$w_2 + \Delta$	$w_3 + \Delta$	$w_4 + \Delta$
Descenso	$w_1 - \Delta$	$w_2 - \Delta$	$w_3 - \Delta$	$w_4 - \Delta$
Pitch (adelante)	$w_1 - \Delta$	w_2	$w_3 + \Delta$	w_4
Pitch (atrás)	$w_1 + \Delta$	w_2	$w_3 - \Delta$	w_4
Roll (derecha)	w_1	$w_2 - \Delta$	w_3	$w_4 + \Delta$
Roll (izquierda)	w_1	$w_2 + \Delta$	w_3	$w_4 - \Delta$
Yaw (horario)	$w_1 - \Delta$	$w_2 + \Delta$	$w_3 - \Delta$	$w_4 + \Delta$
Yaw (anti horario)	$w_1 + \Delta$	$w_2 - \Delta$	$w_3 + \Delta$	$w_4 - \Delta$

Tabla 2
Secuencia de empuje para la configuración en cruz (x).

Movimiento	Rotor 1	Rotor 2	Rotor 3	Rotor 4
Ascenso	$w_1 + \Delta$	$w_2 + \Delta$	$w_3 + \Delta$	$w_4 + \Delta$
Descenso	$w_1 - \Delta$	$w_2 - \Delta$	$w_3 - \Delta$	$w_4 - \Delta$
Pitch (adelante)	$w_1 - \Delta$	$w_2 - \Delta$	$w_3 + \Delta$	$w_4 + \Delta$
Pitch (atrás)	$w_1 + \Delta$	$w_2 + \Delta$	$w_3 - \Delta$	$w_4 - \Delta$
Roll (derecha)	$w_1 + \Delta$	$w_2 - \Delta$	$w_3 - \Delta$	$w_4 + \Delta$
Roll (izquierda)	$w_1 - \Delta$	$w_2 + \Delta$	$w_3 + \Delta$	$w_4 - \Delta$
Yaw (horario)	$w_1 - \Delta$	$w_2 + \Delta$	$w_3 - \Delta$	$w_4 + \Delta$
Yaw (anti horario)	$w_1 + \Delta$	$w_2 - \Delta$	$w_3 + \Delta$	$w_4 - \Delta$

2.2 Captura de Imágenes de la cámara

Los UAV's en la actualidad se encuentran dotados de múltiples sensores para su funcionamiento entre los cuales un dispositivo de visión (una cámara) ha tenido gran acogida en los últimos años (Leos Monroy, 2014) puesto que proporciona una mejor percepción del ambiente con un peso ligero y pequeñas dimensiones, de manera que el UAV tiene la capacidad de posicionarse en el espacio y recopilar grandes cantidades de información rápida y eficientemente.

2.2.1 Representación de una imagen

Una imagen se puede considerar, como objeto de estudio para este trabajo, como la representación en una matriz bidimensional de valores de intensidad de luz de una escena en el mundo tridimensional (Ma, Kosecka, Soatto, & Sastry, 2003). De esta manera se pueden plasmar las características del entorno en una región compacta de dos dimensiones con valores reales positivos como se puede observar en la Figura 7

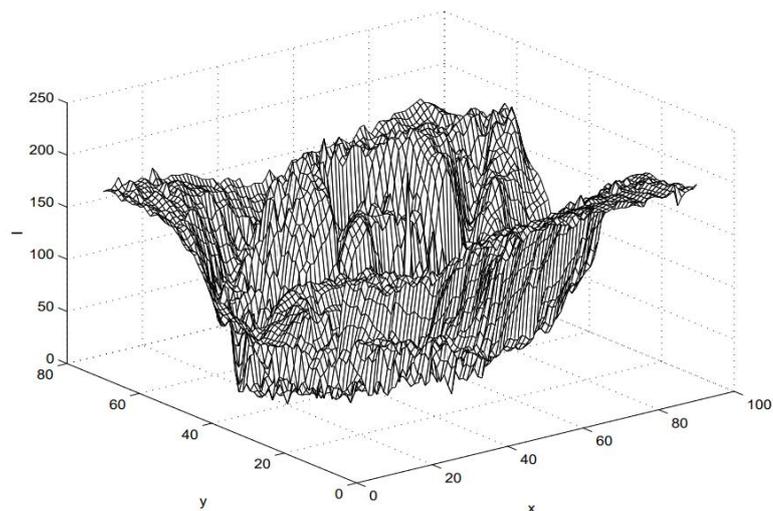


Figura 7. Una imagen I representada como una superficie bidimensional.

Fuente: (Ma et al., 2003)

Para más información acerca del proceso de adquisición de una imagen, los efectos que intervienen en la misma, su reflectancia, refractancia y otros se puede dirigir a (Ma et al., 2003).

2.2.2 Formación de una imagen

En primer lugar, se determina la posición geométrica en el plano de la imagen a partir de la proyección de un punto de la escena. Después se determina el brillo de un punto en el plano de la imagen de acuerdo a las propiedades de la superficie y su iluminación (Forsyth & Ponce, 2012; Viala, 2008).

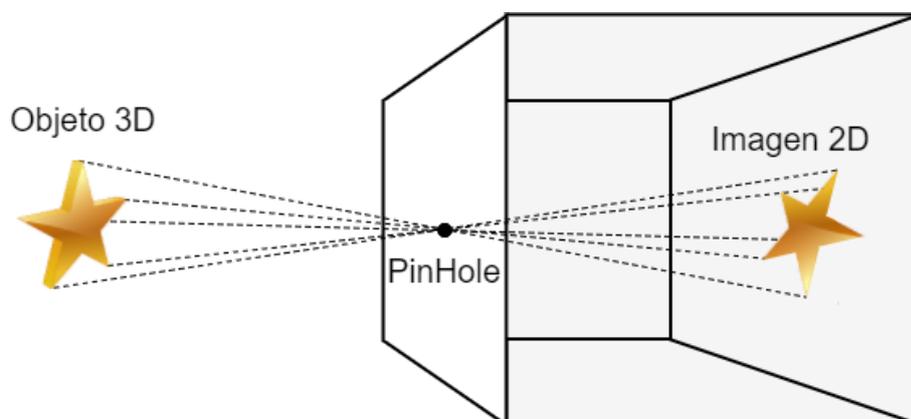


Figura 8. Formación de la imagen con el modelo de Pin-Hole.

Fuente: (Viala, 2008)

El proceso de formación de una imagen plasma una escena 3D en una superficie 2D, una de las más sencillas representaciones de esto es la que se presenta en la Figura 8 donde un pequeño agujero en una caja con dos pantallas paralelas permite el paso de un solo fotón de luz emitido de manera que el objeto en la escena se ve reflejado en la otra pantalla como una imagen invertida del mismo captado por un elemento fotosensible (Forsyth & Ponce, 2012; Viala, 2008).

2.2.3 Modelo de Pin-hole de la cámara

El modelo de cámara de Pin-hole es un modelo proyectivo que permite la representación de una escena perteneciente al mundo (en tres dimensiones) sobre una imagen (en dos dimensiones). El funcionamiento de este modelo se basa en el modelo de la Figura 9 donde se representa de manera sencilla los componentes del mismo.

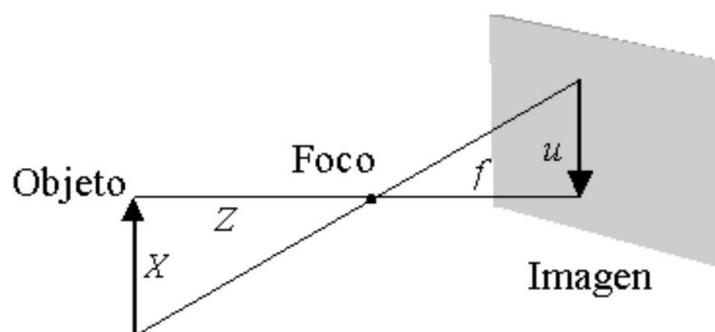


Figura 9. Modelo de Pin-hole.

Fuente: (González, 2003)

Éste modelo es el más sencillo para representar de manera simplificada las características 3D del mundo exterior a la cámara en una imagen de 2D. Para la adquisición se considera que los haces de luz incidentes de la escena exterior (en tres dimensiones) pasan por un solo punto central llamado centro óptico (Platero, 2009) estos haces de luz correspondientes al espacio Euclídeo tras haber pasado por el centro óptico llegan a un plano de imagen que está separado una distancia “f” (distancia focal) del centro óptico.

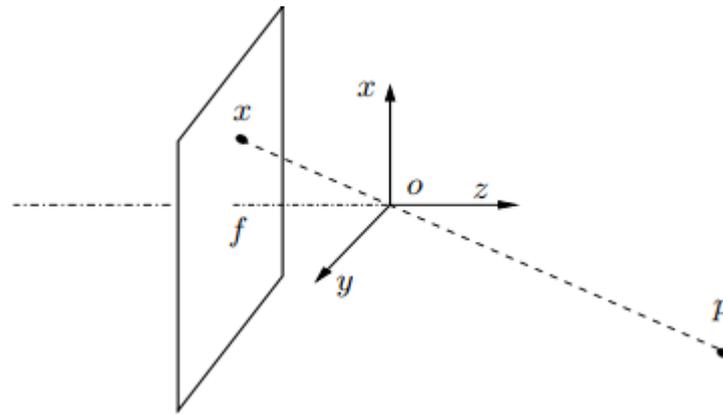


Figura 10. Proyección de un Punto P de la escena 3D en un plano de imagen 2D pasando a través del centro óptico y una distancia focal.

Fuente: (Ma et al., 2003)

La matriz de Proyección de este modelo se puede representar como en (González, 2003), ecuación (3):

$$m \doteq P \cdot M \quad (3)$$

Donde “M” (5) es el vector de coordenadas del punto de la escena 3D, “P” la matriz de proyección con dimensiones de 3x4 y “m” (5) el vector de coordenadas del punto proyectado en la imagen transformada a píxeles y se expresan como en la Figura 11 (González, 2003):

$$M = [X_w, Y_w, Z_w, 1]^t \quad (4)$$

$$m = [u, v, 1]^t \quad (5)$$

El símbolo igual de la ecuación (3) indica la equivalencia de ambos factores salvo un factor de escala de proporcionalidad λ , dando como resultado $\lambda m = P \cdot M$.

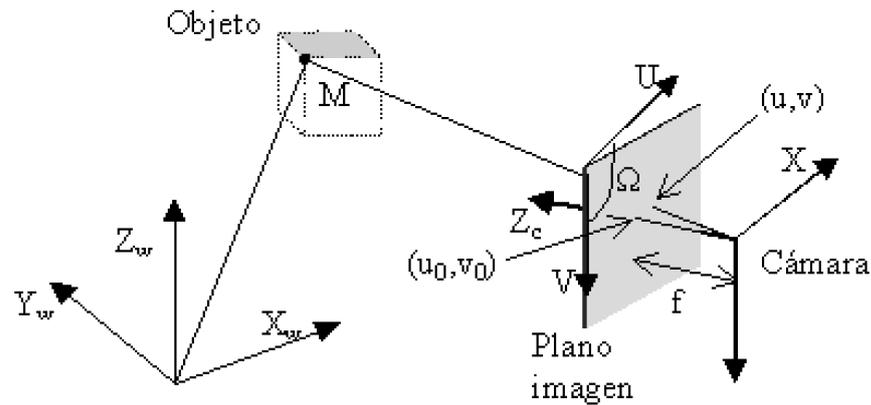


Figura 11. Modelo de cámara. Proyección de puntos en la imagen.

Fuente: (González, 2003)

De esta manera al tener estos tres planos de referencia (Objeto en el mundo, imagen y cámara) se deben realizar los siguientes pasos para convertir finalmente la escena de coordenadas 3D en 2D (González, 2003):

- En primer lugar se pasa las coordenadas 3D de la escena al plano de referencia 3D de la cámara mediante una traslación y rotación de los puntos en la misma (Kühling, 2017), ecuación (6).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (6)$$

De manera simplificada:

$$M_c = R \cdot M \cdot T \quad (7)$$

Donde R es la matriz de rotación, T la de traslación y M el punto de la escena. (R debe ser Ortonormal).

- La proyección sobre el plano imagen con el modelo Pin-hole responde a la ecuación (8) (Jáuregui et al., 2015):

$$u_p = f \frac{X_c}{Z_c} \quad v_p = f \frac{Y_c}{Z_c} \quad (8)$$

Con f como distancia focal. El modelo de Pin-hole denotaría una $-f$ como representación de la imagen invertida respecto a la escena, no obstante eliminar

el signo elimina la inversión de la imagen puesto que si $[u, v]$ representan las coordenadas de la imagen invertida entonces $[-u, -v]$ representaría la imagen en sentido correcto, así se elimina el signo de f logrando obtener la imagen en el sentido correcto (Ma et al., 2003) (Kühling, 2017).

- Posteriormente se agrega al modelo los parámetros de distorsiones del lente de la cámara, entre los cuales se encuentran la distorsión radial y la tangencial, ecuaciones (9)(10).

$$u_d = u_p + u_r + u_t \quad (9)$$

$$v_d = v_p + v_r + v_t \quad (10)$$

- Finalmente se realiza la conversión de unidades (milímetros-píxeles) y se desplaza el origen al centro del eje óptico (11)(12).

$$u = d_x s_x u_d + u_0 \quad (11)$$

$$v = d_y v_d + v_0 \quad (12)$$

Los parámetros d_x, d_y son factores de conversión, s_x un factor de proporción del píxel y u_0, v_0 las coordenadas del centro del eje óptico.

2.3 Caracterización de las imágenes

Las imágenes capturadas por una cámara y estimadas a un modelo 2D que representa el entorno 3D de la escena captada permiten tener una noción más completa que únicamente con sensores inerciales para el espectador. El ojo humano puede extraer toda la información necesaria, por otro lado en el procesamiento computacional para realizar tareas como reconocimiento de patrones, estimación de movimiento y demás se necesitan extraer los parámetros característicos que incluyen estas imágenes y así poder discriminarlos en la escena para poder realizar acciones pertinentes.

Las imágenes de computadora se pueden representar como en (Nixon & Aguado, 2012), como una matriz de dos dimensiones de píxeles donde cada píxel tiene un valor de intensidad que usualmente es de 8 bits, se ajusta a la escala de $2^m - 1$ donde $m=8$ (es decir de 0-255 de negro a blanco en la escala de grises), este valor es proporcional

a la intensidad del punto en la escena 3D. Las imágenes a color llevan un modelo parecido, la diferencia es que se usa más de un canal para representar a cada pixel; uno de los modelos que generalmente usan es el RGB donde cada pixel tiene un valor en rojo, verde y azul en una escala que también suele ser de 8 bits representando así el color real de la imagen en la escena.

Existen múltiples operaciones que se pueden realizar a imágenes ya sea pixel por pixel o al conjunto frame en sí, como la aplicación de un cambio de contraste, umbral de contraste, operaciones estadísticas, convoluciones, filtros gaussianos, erosión, dilación y demás (Nixon & Aguado, 2012).

2.3.1 Métodos de caracterización de imágenes

Los métodos para representación de imágenes según sus características están divididos entre globales y locales (Figura 12) (Hassaballah, Abdelmgeid, & Alshazly, 2016).

La representación global de la imagen incluye a cada pixel describiendo toda la imagen en el proceso; el color, textura o forma de la imagen pueden ser incluidos como valores para el vector que la describe de manera global, tiene la ventaja de ser un conjunto más compacto y rápido de procesar. Este método de representación no es muy útil para la discriminación de patrones específicos o cambios de estructura de una imagen, como puntos, bordes o regiones que difieren de su entorno por textura, color o intensidad.

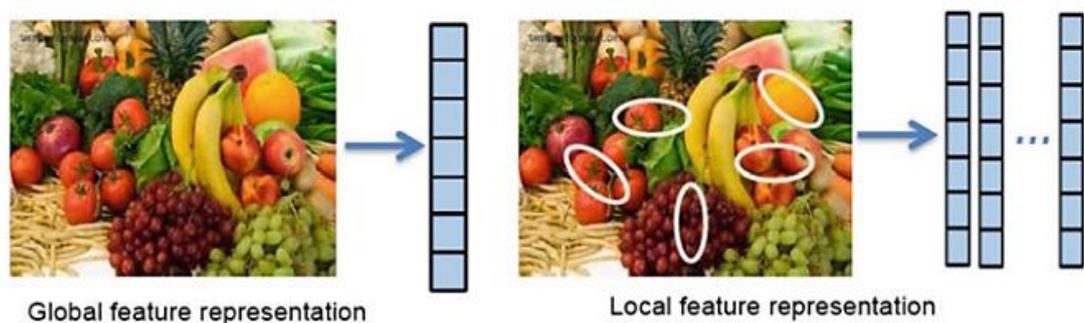


Figura 12. Representación Global y Local de las características de una imagen.

Fuente: (Hassaballah et al., 2016)

Las características locales de una imagen se enfocan en las regiones destacadas de la misma, llamadas regiones de interés (Figura 13), estas regiones se caracterizan por tener peculiaridades en su estructura, iluminación y textura además de ser invariantes a las transformaciones significativas, insensibles al desorden y la oclusión.

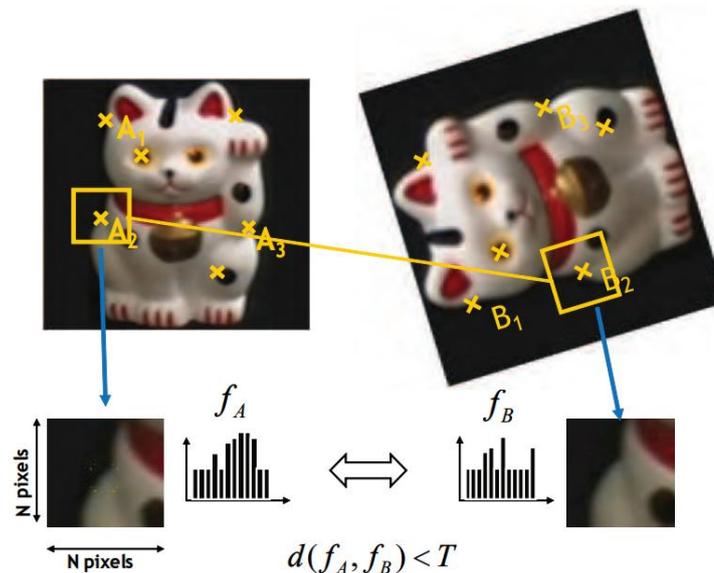


Figura 13. Ejemplo puntos de interés de una imagen.

Fuente: (Grauman & Leibe, 2011)

Mediante el uso de características locales se logra la reducción de los recursos necesarios para representar una imagen, haciéndolo en una pequeña cantidad de datos significativos e invariables (Saber & Sayem, 2016) proporcionando la posibilidad de emparejar estructuras locales de imágenes de manera más eficiente, estos conjuntos de datos deben cumplir con ser suficientes, compactos, invariantes y congruentes (Hassaballah et al., 2016).

2.3.1.1 Tipos de Características locales

Las características distintivas locales dentro una imagen pueden ser bordes o contornos, puntos de interés (o esquinas como son llamados popularmente) y blobs. Las definiciones de estos parámetros según (Saber & Sayem, 2016) son:

Bordes.- los puntos de la imagen donde la dirección del gradiente en un sentido es grande de manera que combinados pueden formar límites, regiones o líneas.

Esquina (Punto de interés).- es una característica puntual que tiene su posición bien definida y no necesariamente son esquinas, es decir la intersección de dos bordes, sino también puede ser un punto aislado con gran cantidad de información relevante a su alrededor como mínimos o máximos de filtros locales aplicados a la imagen.

Blobs (Manchas).- estas son características de mayor dimensión en una imagen, no pueden ser detectadas por la detección de esquinas o bordes puesto que son demasiado lisas, con poca intensidad o falta de cambios bruscos en ellas.

El objetivo de este trabajo es detectar características lo suficientemente relevantes para discriminar objetos definidos en una imagen de manera que se enfocará a la detección de características locales con Puntos de interés por ser robustos en repetitividad y cambios visualización.

2.3.2 Proceso de caracterización de una imagen

En este proceso se comienza detectando las características de una imagen, estos deben ser los puntos de interés o regiones distintivas, le sigue el proceso de codificación de la información donde se determina la información espacial de las regiones representadas matemáticamente por el pixel tomado como característico, es decir, se le da un vector que representa su aspecto visual y lo diferencia de sus vecinos. Luego se procede a comparar los vectores descriptores de una imagen I_0 con los de una imagen I_1 para obtener los puntos característicos coincidentes.

2.3.2.1 Detección de puntos de interés

La detección de puntos de interés debe realizarse de manera rigurosa, tal que los puntos localizados tengan características invariantes a los cambios morfológicos de la imagen, cambios de visualización y ante el ruido (Saber & Sayem, 2016). Los extractores de características deben cumplir varios criterios (Grauman & Leibe, 2011) (Hassaballah et al., 2016):

- Repetitividad y precisión en la extracción (Robustez), localizando en su mayoría características similares en imágenes que tienen el mismo objeto con respecto a su ubicación en pixeles. Además deben poder localizarse los mismos puntos a pesar de los cambios de visualización y presencia de ruido

- Discriminación, las características del punto y sus alrededores con respecto a otros puntos son distintivas y permiten distinguirlos de manera confiable.
- Eficiencia, la detección debe realizarse mediante algoritmos que permitan la rapidez y precisión de la misma y así poder usarlos en tiempo real para alguna aplicación.
- Cantidad, todas las características de la imagen deben poder ser localizadas en los puntos encontrados y con ellas se debe poder representar a toda la imagen de una manera compacta.

Existen varias propuestas para la detección rápida y confiable de características locales en la imagen, entre las cuales destacan los métodos de extracción por gradiente y variación de intensidad.

Los detectores de puntos de interés por lo regular usan máximos y mínimos (gradientes) luego de haber aplicado filtros gaussianos para reducir el ruido en la imagen (Krig, 2014), no obstante pueden realizarse muchas otras operaciones en el conjunto local de píxeles con el fin de obtener diferentes grados de invariancia y robustez en la detección, como por ejemplo la magnitud y dirección del gradiente, el laplaciano, la matriz Hessiana y sus variaciones, así como las propiedades propias (eigen properties) en valores, vectores y espacios (Krig, 2014), estas operaciones se ajustan a los diferentes métodos de detección que se utilicen.

Algunos de los métodos más conocidos y que servirán como base para los algoritmos usados en siguientes capítulos son los siguientes:

- Método de Harris.- uno de los métodos más básicos en la detección de bordes y esquinas cuyos conceptos son bases sólidas para los algoritmos actuales.
- Método de Diferencia de Gaussianas.- método que permite gran invariancia al escalado y es utilizado por el algoritmo SIFT.
- Método de la matriz Hessiana.- método que presenta gran rapidez en la detección y es utilizado por el algoritmo SURF.
- Método FAST.- método binario de detección usado por el algoritmo ORB.

a. Métodos de Harris

(Krig, 2014) Los detectores de Harris combinan la detección de bordes y esquinas, optan por encontrar los cambios de dirección más rápidos y lentos para las características de orientación mediante la variación de autocorrelación (variación de intensidad) en todas las direcciones dando como resultado una alta invariancia rotacional y robustez ante cambios de intensidad. La matriz de cambios de dirección en cada pixel de la imagen se expresa como en la ecuación (13) de (Hassaballah et al., 2016):

$$M(x, y) = \sum_{u,v} w(u, v) * \begin{bmatrix} I_x^2(x, y) & I_x I_y(x, y) \\ I_x I_y(x, y) & I_y^2(x, y) \end{bmatrix} \quad (13)$$

Los parámetros correspondientes de la ecuación (13) son: I_x , I_y los gradientes de la imagen en las direcciones x, y; luego se usa una ventana de ponderación sobre el área (u,v) al usar una gaussiana se tendrá la suma de los pixeles de una región local circular (Grauman & Leibe, 2011), posteriormente para obtener el punto de interés se deben calcular los valores propios de la matriz M para cada pixel que luego se comparan con cierto parámetro de ajuste “K” dando los valores que denota aplicar la ecuación (14):

$$C(x, y) = \det(M) - K(\text{trace}(M))^2 \quad (14)$$

Donde

$$\det(M) = \lambda_1 * \lambda_2, y \text{ trace}(M) = \lambda_1 + \lambda_2 \quad (15)$$

El tipo de punto encontrado se define con respecto al criterio de la Figura 14.

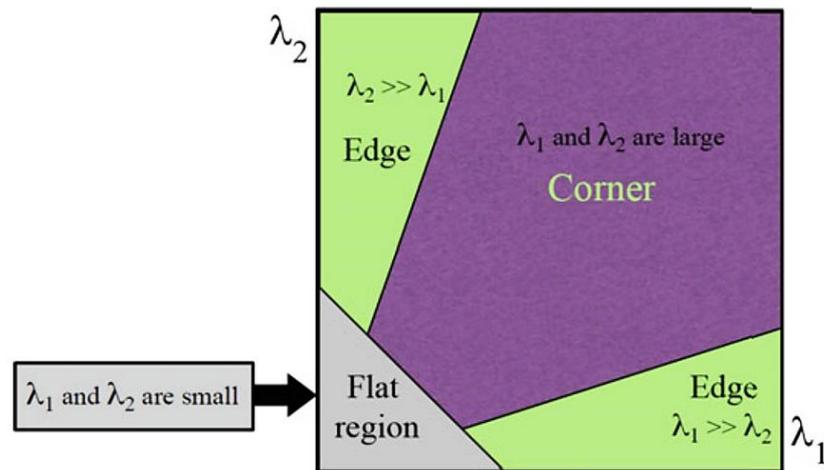


Figura 14. Clasificación de los puntos en la imagen basada en los valores propios de la matriz de autocorrelación M.

Fuente: (Hassaballah et al., 2016)

b. Método de Diferencia de Gaussianas

El método de diferencia de Gaussianas (DOG) es un aproximado del Laplaciano de Gaussianos y se usa para detectar los extremos locales del espacio escala además se puede calcular como una diferencia de niveles de escala adyacentes de una pirámide Gaussiana separados un factor σ (Hassaballah et al., 2016), lo que se puede apreciar en la Figura 15 donde x, y son las ubicaciones de cada pixel y σ la octava o ventana sometida a un nivel de escala correspondiente:

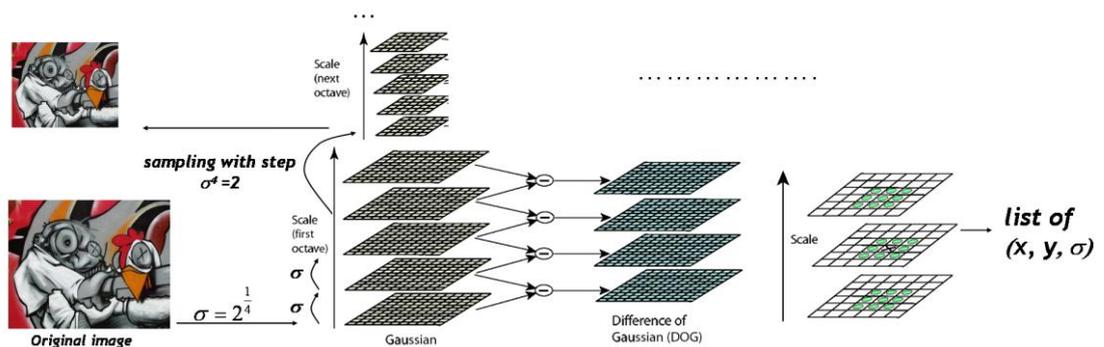


Figura 15. Búsqueda de los extremos en 3D de espacio escala con DOG.

Fuente: (Tuytelaars & Mikolajczyk, 2008)

La función se puede obtener de la siguiente manera (Grauman & Leibe, 2011) ecuación (16):

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (16)$$

Para obtener los puntos de interés se comparan las ubicaciones de los extremos a lo largo de la función $D(x, \sigma)$ con cada punto de su vecindad de 8 de la misma escala y 9 más cercanos en cada nivel adyacente. Las regiones que se mantienen son las que caen en cierto umbral t y cuyas escalas están en un rango específico.

c. Método de la Matriz Hessiana

Este método detecta regiones de interés de un conjunto de imágenes de múltiples escalas donde se calculan las derivadas parciales Gaussianas de segundo orden de intensidad de imagen $I(x, y)$ (por el ruido introducido), a esta se le denomina la matriz de Hesse (17), el objetivo de esto es buscar las regiones de mayor cambio en dos direcciones ortogonales (Grauman & Leibe, 2011) (Krig, 2014).

$$H(x, y, \sigma) = \begin{bmatrix} I_{xx}(x, y, \sigma) & I_{xy}(x, y, \sigma) \\ I_{xy}(x, y, \sigma) & I_{yy}(x, y, \sigma) \end{bmatrix} \quad (17)$$

El determinante de esta matriz está en un máximo de manera que funciona en los extremos locales (18).

$$\det(H) = I_{xx}I_{yy} - I_{xy}^2 \quad (18)$$

Entonces este método usa el determinante en múltiples escalas para la localización espacial en píxeles comparando con sus vecinos en el cuadrante de 3×3 y devuelve los valores que superan un umbral predefinido (Hassaballah et al., 2016).

d. Método FAST

Del inglés (Features from Accelerated Segment Test) FAST es un algoritmo de detección de puntos de interés binario que basa su funcionamiento en comparaciones del nivel de intensidad sucesivos entre los píxeles candidatos.

FAST realiza las comparaciones del nivel de intensidad del píxel p candidato con un círculo de Bresenham de radio r , p es el centro del círculo, para decidir si el

candidato es un punto de interés se evalúa el nivel de intensidad de p como I_p con un nivel de umbral t , de manera que cada pixel a ser comparado es mayor en intensidad si su brillo es mayor a $I_p + t$ o es menor en intensidad si es menor que $I_p - t$. En la Figura 16 se puede apreciar el método FAST para comparar el pixel central con un círculo de $r = 3$ (Hassaballah et al., 2016).

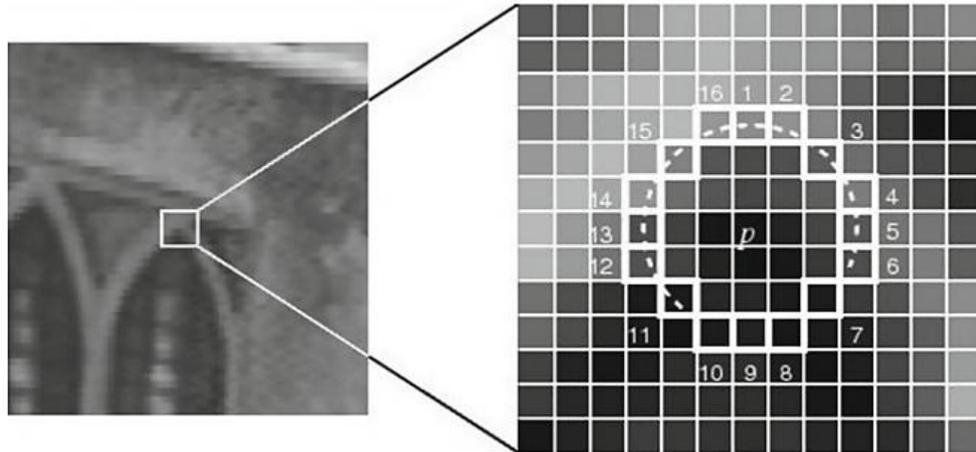


Figura 16. Comparación de intensidad de un pixel mediante FAST en un círculo de radio 3.

Fuente: (E Rosten & Drummond, 2005) citado en (Hassaballah et al., 2016)

Como en la Figura 16 al comparar estos píxeles se determina que el píxel en cuestión es un punto de interés si “n” cantidad de píxeles consecutivos de la comparación son todos mayores o menores en intensidad que el candidato más un umbral t , por lo regular el radio es $r = 3$ y la cantidad de píxeles $n = 12$. Una propuesta del algoritmo de detección es la prueba de alta velocidad donde solo se comparan los píxeles 1, 9, 5 y 13, si es un punto de interés al menos tres de estos píxeles deben ser más brillantes u oscuros que el píxel central más el umbral.

Al almacenar las posiciones de cada píxel con respecto a su brillo en un vector de 16 ubicaciones se puede obtener una característica invariante a la rotación (Krig, 2014) por su naturaleza circular, para mejorar el rendimiento de la prueba de velocidad se utiliza un algoritmo de aprendizaje automático (ID3) (Edward Rosten & Drummond, 2006) para una localización precisa de las características.

2.3.2.2 Descripción de puntos de interés

Un vez que se han detectado los puntos característicos de la imagen se debe realizar el proceso de codificación de la información, se describe la información espacial de las regiones representadas matemáticamente por el pixel tomado como característico, es decir, se le da un vector que representa su aspecto visual, el de su región circundante y permite la coincidencia discriminativa (Grauman & Leibe, 2011).

Para poder realizar la descripción de características en un conjunto discriminativo e insensible a las deformaciones locales de la imagen se usan los parámetros obtenidos en la detección de puntos de interés, que se detectan junto con su posición, escala y orientación (Fan, Wang, & Wu, 2015; Hassaballah et al., 2016). Posteriormente estos cálculos se representan como un conjunto de números de manera vectorial los cuales basan su filosofía en cuatro propiedades fundamentales (Nixon & Aguado, 2012):

- Definir un conjunto completo, si las propiedades visuales de dos objetos son iguales entonces deben tener los mismos descriptores.
- Tener congruencia, si los descriptores de dos objetos son similares entonces deben ser parecidos.
- Tener propiedades invariantes, para que se puedan detectar descriptores parecidos a pesar de los cambios en el punto de vista de las imágenes coincidentes.
- Ser un conjunto compacto, se debe representar la esencia de un objeto con pocos datos diferenciables a otros objetos

Los descriptores de puntos de interés suelen personalizarse a medida que el algoritmo lo requiera, no obstante varios enfoques se usan como métodos (Krig, 2014).

Descriptores locales binarios.- estos descriptores representa las características como vectores de bits binarios de la comparación de pixeles. De este enfoque salen las comparaciones de vecindario donde se comparan las características del pixel y sus vecinos, los histogramas de composición para describir las características de textura, concatenación y otros.

Descriptores de espectro.- estos descriptores necesitan más coste computacional que los binarios, en muchos casos usan información de gradientes, histogramas, momentos estadísticos y otras; tienen características de invariabilidad que en ocasiones suelen ser mejores que los descriptores binarios que carecen de invariabilidad al escalamiento.

Además de estos descriptores existen otros que no serán abordados en este estudio como los de Espacio base, los de forma de polígono y los volumétricos de 3D y 4D.

2.3.2.3 Emparejamiento de coincidencias

En consecuencia a la obtención de los descriptores de cada punto de interés encontrado en la imagen, el proceso posterior es el emparejamiento, es decir encontrar correspondencias entre puntos de fotogramas. Se obtienen comparando los descriptores de cada uno de los fotogramas como una distancia vectorial con un error, la correspondencia correcta es la muestra que difiere menos entre los puntos de los fotogramas asociados.

El proceso de emparejamiento empieza con la medida de la distancia entre los vectores descriptores, esto se puede realizar como se muestra en (Hassaballah et al., 2016) con la resta de los mismos para cada vector asociado al punto de interés de la imagen 1 y de la imagen 2, para ejemplificar se tomaran como p, q respectivamente y se representa según la ecuación (19).

$$d_k(p, q) = |\phi_k(p) - \phi_k(q)| \quad (19)$$

Donde $d_k(p, q)$ es la distancia de los k vectores descriptores entre p y q , $\phi_k(p)$ y $\phi_k(q)$ representan los k vectores descriptores de p y q respectivamente. Si Q es el conjunto de puntos de interés de la imagen 2, tal que, $Q = \{q_1, q_2, \dots, q_N\}$ uno de los métodos para emparejar es evaluar la ecuación anterior con p y todo el conjunto Q para luego ordenar los resultados en función de la distancia d_k y obtener de tal manera los puntos del conjunto Q que tienen menor distancia con p . En este método se evalúan dos criterios para aceptar la coincidencia en el par de puntos (p, q) según (Hassaballah et al., 2016):

- (i) p es la mejor coincidencia para q en relación con todos los otros puntos en la primera imagen y
- (ii) q es la mejor coincidencia para p en relación con todos los otros puntos en la segunda imagen.

Este es el método más sencillo que permite comprender el concepto de emparejamiento, no obstante existen métodos específicos con muy buen rendimiento para realizar este proceso, uno de los cuales se llama el del vecino más cercano que empiezan con una medida de la distancia euclidiana en variables continuas o la métrica de Hamming en discretas (estas métricas se explican más adelante en la aplicación de los métodos) entre los vectores descriptores, posteriormente encuentra los puntos no exactos que se acercan más al punto de entrenamiento. Se puede especificar el número de puntos más cercanos a analizarse y al ser un algoritmo con aprendizaje no paramétrico mejora el rendimiento y la rapidez de cálculo, la descripción completa y más métodos se pueden revisar en (Stuart, 2010).

2.3.3 Algoritmos de detección y descripción de puntos de interés

Una de las principales ventajas a la hora de utilizar los algoritmos de detección de puntos de interés existentes es el bajo coste computacional de los mismos (Aguilar & Angulo, 2012), existen múltiples algoritmos que permiten realizar la detección de puntos de interés (Krig, 2014); los detectores de Harris que tienen una alta invariancia rotacional y robustez ante cambios de intensidad por ejemplo, o los de FAST que al ser binarios permiten una rapidez mayor junto con una invariancia rotacional o los de múltiple escala como DOG Diferencia de Gaussianos que permite invariancia a la escala, no obstante algunos algoritmos que han adquirido gran popularidad por su buen desempeño tienen sus propias modificaciones a los métodos convencionales e incluyen detectores y descriptores de puntos de interés como lo son SIFT (Scale Invariant Feature Transform (D.G. Lowe, 1999)), SURF (Speeded Up Robust Features (Bay, Tuytelaars, & Van Gool, 2006)) y ORB (Oriented FAST and Rotated BRIEF (Rublee, Rabaud, Konolige, & Bradski, 2011)).

Estos métodos han sido ampliamente usados en el emparejamiento de imágenes para múltiples aplicaciones por la comunidad científica (Işık & Özkan, 2014) junto con BRISK, BRIEF, HARRIS, FAST y MSER comparándolos y también comparando

las combinaciones de estos algoritmos que dan mejor rendimiento en el emparejamiento por puntos característicos. En este estudio se van a abordar únicamente tres de estos algoritmos SIFT, SURF y ORB.

2.3.3.1 Algoritmo SIFT

SIFT es un algoritmo de alta eficiencia presentado por David Lowe (D.G. Lowe, 1999), característicamente cuenta con una alta invariancia a los cambios de traslación, rotación y escalamiento permitiendo una buena detección de puntos de interés en ambientes ruidosos como es el caso de un dron que realiza la captura de video en el aire. Consta de cuatro etapas en las que se evidencia como objetivo minimizar el costo computacional al realizar operaciones grandes solo en ubicaciones específicas de un filtrado en cascada (Cheng-Yi, Chen and Shih-Hong, 2012; Ilbay, Córdova, Raura, & Gualotuña, 2013; Saber & Sayem, 2016).

- **Detección de los extremos en la escala:** para realizar esta operación primero se generan imágenes a diferentes dimensiones y escalas a partir de la imagen original, posteriormente para identificar los posibles puntos de interés se aplica una diferencia de Gaussianas como se vio en una sección anterior en las imágenes a diferentes niveles de desenfoque, esto hace a los puntos de interés invariantes a los cambios de orientación y escalado; se calcula mediante la ecuación (16) que posteriormente es analizada en su vecindario de 8 vecinos en la misma escala y 9 en las escalas adyacentes, al analizar la función en cada punto se elige el mismo si es el máximo de sus 26 vecinos como se evidencia en la , el documento original sugiere 4 octavas y 5 niveles de desenfoque como óptimas.

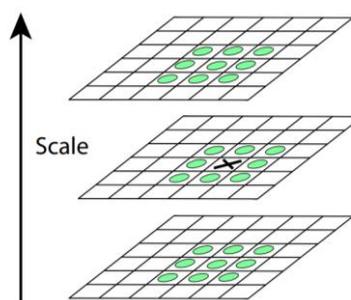


Figura 17. Comparación del punto con sus 26 vecinos más cercanos.

Fuente: (David G Lowe, 2004)

- **Localización de keypoints:** se eligen en base a medidas de estabilidad donde se rechazan los puntos de bajo contraste local mediante un filtrado con un umbral definido que permite descartarlos. También se eliminan los puntos que se encuentran localizados en los bordes mediante el concepto introducido por los valores propios de la matriz de Harris donde si un valor propio es muy grande y el otro muy pequeño el pixel en cuestión es de borde.
- **Asignación de la orientación:** la característica de orientación de cada punto de interés se calcula por un análisis del gradiente de imagen local.
- **Descriptor del punto clave:** se miden gradientes locales alrededor del punto mediante un histograma 3D de ubicaciones y orientaciones en una ventana 16×16 , valores que son resumidos y almacenados en una matriz de descriptores 4×4 proporcionando un descriptor de 128 elementos aunque hay otras disposiciones que dependiendo de la matriz de muestreo cambian, luego son normalizados para una representación que permita distorsiones y cambios de iluminación en forma local. En la Figura 18 se muestra el ejemplo para una ventana de 8×8 con descriptores de 2×2 .

$$M_{SIFT}(x, y) = \sqrt{((L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2)^2} \quad (20)$$

$$\theta_{SIFT}(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (21)$$

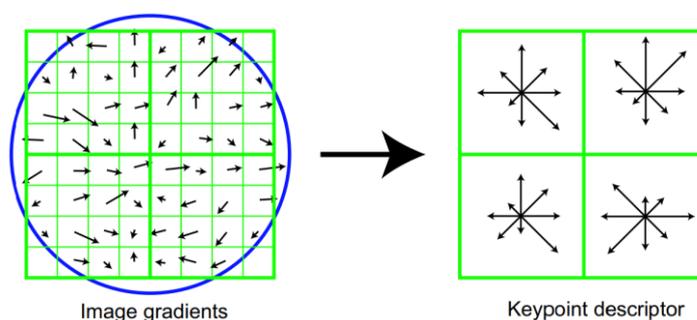


Figura 18. Ejemplo de descriptor del punto de interés en una ventana de 8×8 que genera un descriptor de 2×2 .

Fuente: (David G Lowe, 2004)

La descripción completa del método se puede apreciar en (Fan et al., 2015).

2.3.3.2 Algoritmo SURF

Este algoritmo se presentó basado en la premisa de que los detectores basados en la matriz Hessiana son más estables y repetibles que los basados en Harris y al usar el determinante de H son más rápidos que usar DOG. El detector de SURF se basa en el determinante de la matriz de Hesse la cual se evidencia en la ecuación (17), usando este para la ubicación y escala, además en lugar de usar filtros Gaussianos para el espacio escala optan por filtros de caja simples 2D en imágenes integrales independientemente del tamaño de la imagen, un filtro de 9×9 de escala $s = 1.2$ y sirve como una aproximación del método de derivadas gaussianas de segundo orden con $\sigma = 1.2$, y sirve como capa inicial para seguir obteniendo las demás capas con máscaras del filtro gradualmente mayores (Bay, Ess, Tuytelaars, & Van Gool, 2008).

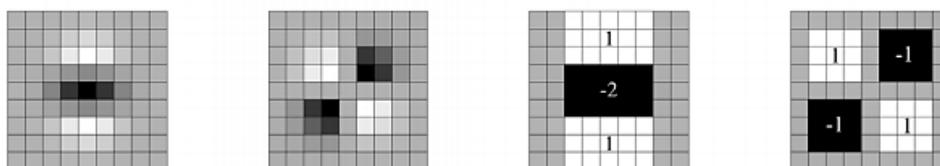


Figura 19. En la mitad izquierda derivadas parciales Gaussianas de segundo orden en dirección y, xy. En la mitad derecha el uso de filtros de caja; la región negra es 0.

Fuente: (Bay et al., 2008)

El determinante aproximado que usa este método se puede observar (22).

$$\det(H_{aprox} = D_{xx}D_{yy} - (wD_{xy})^2) \quad (22)$$

w Es un peso relativo que equilibra la expresión del determinante de Hesse y sirve para la respuesta del filtro, finalmente para encontrar los puntos de interés se toman los máximos locales aplicando una supresión no máxima en el vecindario de $3 \times 3 \times 3$ y usando una interpolación cuadrática del espacio escala de los mismos haciéndolos estables en espacio y escala.

Para la descripción SURF describe la distribución de intensidad del vecindario alrededor del punto de interés, basado en la distribución de las respuestas de la función Wavelets de Haar Figura 20 de primer orden en dirección x e y (Bay et al., 2008). El método consiste primero en fijar una orientación reproducible basada en el área circundante al punto de interés, para luego extraer el descriptor de una región cuadrada alineada a la orientación fijada anteriormente.

La orientación se asigna calculando la respuesta de los Wavelets de Haar en el vecindario circular de radio $6s$, donde s es la escala de detección del punto de interés, posteriormente se aplica una ponderación Gaussiana de $\sigma = 2s$ centrada en el punto de interés representando estas respuestas en un plano que al ser sumados en fragmentos de $\pi/3$ se obtienen las orientaciones dominantes Figura 21, luego se suman las respuestas horizontales y verticales dando como resultado un vector de orientación local, el vector más grande es el que le da la orientación al punto de interés con invariancia a la rotación.



Figura 20. Filtros de Haar para la respuesta en x , y . La parte negra tiene peso -1 y la blanca $+1$.

Fuente: (Bay et al., 2008)

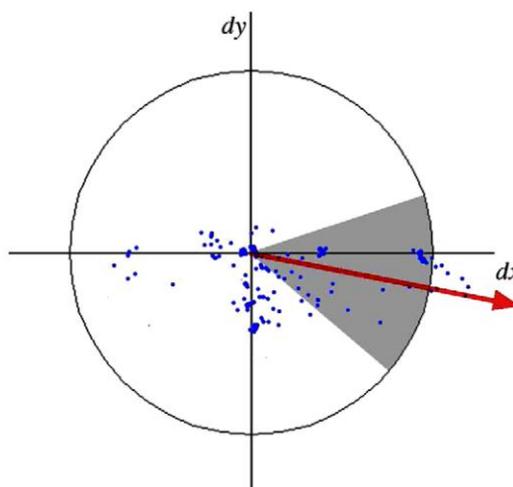


Figura 21. Asignación de la orientación en ventanas de $\pi/3$.

Fuente: (Bay et al., 2008)

El siguiente paso es generar una ventana cuadrada de área $20s$ orientada en dirección a la orientación calculada, luego se divide en subregiones 4×4 calculando en ellas la función de Haar en 5×5 puntos espaciados regularmente al que posteriormente se le aplica una ventana de ponderación Gaussiana para mejorar la exactitud Figura 22. Si la respuesta de la función de Haar en (x, y) es (dx, dy) entonces al sumar las respuestas de la función en cada subregión se obtiene el vector (23):

$$v = (\sum dx, \sum |dx|, \sum dy, \sum |dy|) \quad (23)$$

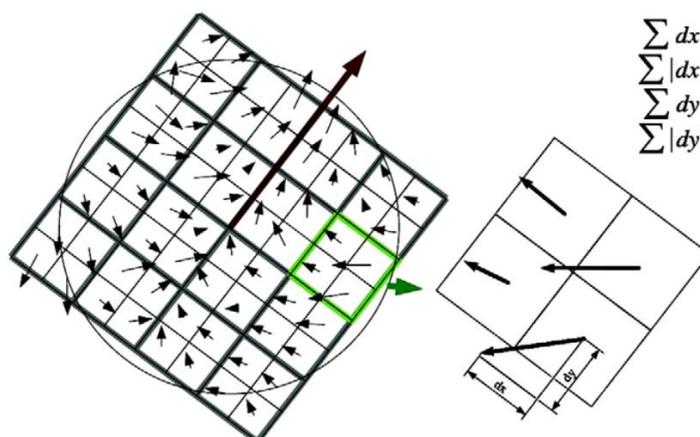


Figura 22. Ilustración a modo de ejemplo del descriptor de SURF con regiones de 4×4 y subregiones de 2×2 y las respuestas al filtro Haar dx, dy .

Fuente: (Bay et al., 2008)

Si se realiza en las cuatro subregiones se obtiene el vector descriptor de $4 \times 4 \times 4 = 64$ dimensiones el cual al ser normalizado es robusto ante deformaciones e invariante al contraste. Estas características hacen que SURF sea más rápido que SIFT.

2.3.3.3 Algoritmo ORB

El algoritmo ORB presentado por (Rublee et al., 2011) tiene el objetivo de mejorar la eficiencia de los algoritmos de detección y descripción de puntos de interés para tener más rendimiento en las aplicaciones comunes de coincidencia de imágenes, ORB

comparado con los otros algoritmos estudiados SIFT y SURF presenta mejores resultados reduciendo el tiempo de procesamiento.

Está basado en el detector de puntos de interés de FAST y los descriptores de BRIEF, estos algoritmos destacan por su rapidez de operación y bajo coste computacional y le dan su nombre al algoritmo ORB (Oriented FAST and Rotated BRIEF), además realiza modificaciones en ambos algoritmos para mejorar su desempeño; se le agrega un componente de orientación a FAST al cual se lo nombra “oFAST” e invariancia rotacional a BRIEF “rBRIEF”, (Zhu, Shen, & Chen, 2016) este proceso se detalla a continuación:

oFAST.- (FAST KeypointOrientation) se utiliza un algoritmo de FAST-9 el cual toma un parámetro de intensidad del pixel y lo compara como pixel central con el círculo de Bresenham de radio $r = 3$, posteriormente si la intensidad de los nueve pixeles consecutivos o más del círculo de Bresenham son mayores o menores al pixel central más un umbral se lo toma como punto de interés, este proceso devuelve medidas grandes en los bordes de modo que se le aplica una medida de esquinas de Harris con un umbral para ordenar los N puntos de interés más aptos para ser tomados. Para añadirle la invariancia al escalado al método se realiza la medida de FAST filtrado por Harris a una pirámide de escala de la imagen en cada nivel.

Posteriormente se le agrega un componente de orientación mediante el centroide de intensidad donde el momento $m_{p,q}$ de un punto de interés “ P ” se muestra en la ecuación (24):

$$m_{p,q} = \sum_{x,y} x^p y^q I(x, y) \quad (24)$$

Donde el centroide “ Q ” de intensidad de los puntos $N(x, y)$ del vecindario cercano en la ventana de radio r correspondientes a la ecuación (24) es:

$$Q = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (25)$$

Finalmente la orientación θ se toma como el vector desde P hasta Q donde P es el origen y se representa en la ecuación (26)(24):

$$\theta = \tan^{-1} \left(\frac{m_{01}}{\frac{m_{00}}{\frac{m_{10}}{m_{00}}}} \right) = \tan^{-1} \frac{m_{01}}{m_{10}} \quad (26)$$

rBRIEF.- (Rotation-AwareBrief) este descriptor realiza simples pruebas binarias entre pixeles, pero no es invariante a la rotación, el añadir la orientación del punto de interés θ al método agrega una invariancia rotacional acorde a θ , el test binario a una ventana p se describe como en la ecuación (27):

$$\tau(p; x, y) = \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases} \quad (27)$$

Donde $p(x)$ es la intensidad del punto “x”, “y” es la distribución gaussiana en las proximidades de x ; los descriptores se obtienen en un vector de n ($n = 256$) test binarios como se aprecia en la ecuación (28):

$$f_n(p) = \sum_{l \leq i \leq n} 2^{i-l} \tau(p; x, y) \quad (28)$$

Para dirigir la matriz S de n pruebas binarias hacia la orientación de cada punto x_i se usa la matriz de rotación R de θ_i , dándole dirección a S de manera que $S_\theta = R_\theta \cdot S$, finalmente el descriptor se representa de la forma:

$$ORB(i) = f_n(p) | (x_i, y_i) \in P_\theta \quad (29)$$

2.4 Transformaciones Geométricas

Uno de los principales enfoques a la hora de encontrar las deformaciones en las imágenes tomadas por una cámara en un lapso de tiempo son los modelos en dos dimensiones (Aguilar & Angulo Bahón, 2013) que pueden abordar parámetros como la traslación, rotación y escalamiento, estos modelos con respecto a su complejidad son: de traslación, de semejanza, afín y proyectivos. Tomando el caso de una cámara que se traslada de un punto $P1 \rightarrow P2, \{P1, P2\} \in \mathbb{R}^3$, la representación matemática de estos movimientos se pueden estimar en base a una concatenación de las transformaciones geométricas de las imágenes tomadas por la misma (Forsyth & Ponce, 2012; Hartley & Zisserman, 2003).

Consecuentemente si se tiene un conjunto de puntos P_1 y P_2 correspondientes a un frame específico y uno posterior al mismo, van a existir un número N de puntos entre

P_1 y P_2 que se ven relacionados entre sí por medio de una transformación geométrica, representada como una matriz de transformación H no singular de dimensiones 3×3 la cual mantiene todas las propiedades proyectivas de los conjuntos de puntos invariables, esto permite estimar el movimiento tridimensional de la cámara mediante un modelo de dos dimensiones homogéneo de conjuntos lineales, los grados de libertad que tienen estos modelos varían respecto a los subgrupos proyectivos que presentan estas transformaciones (Hartley & Zisserman, 2003).

Los modelos paramétricos de movimiento se pueden calcular basándose en las transformaciones geométricas existentes, el tipo de transformaciones a abordarse por ser las más representativas en este trabajo son del modelo no reflectivo, afín y proyectivo.

2.4.1 Transformación de similitud

Esta transformación pertenece al modelo afín con la peculiaridad de ser una isometría que lleva una escala asociada, de manera que este modelo asocia la traslación del conjunto de puntos, una matriz de rotación ortogonal representando un plano de captura paralelo al de la imagen y una escala de separación al mismo, puede ser expresada de la siguiente manera (30):

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (30)$$

- La transformación de similitud se puede obtener desde dos correspondencias de puntos entre frames y tiene cuatro grados de libertad, esta transformación sirve muy bien cuando el plano de transformación es paralelo al de la toma.

2.4.2 Transformación afín

Esta transformación también llamada de afinidad es lineal y no singular con la que se asocian la traslación, escalado, rotación en un eje e inclinación donde la escala (a diferencia de la matriz de similitud) es no isotrópica, teniendo escalas λ_1 y λ_2 asociadas a la rotación en las direcciones x, y respectivamente, esta matriz se representa de la siguiente manera (31):

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (31)$$

O de una manera más sencilla como (32):

$$x' = H_A x = \begin{pmatrix} A & t \\ 0 & 1 \end{pmatrix} x \quad (32)$$

En (32) se puede apreciar que la transformación de afinidad se asocia a una matriz A con una de traslación t , la matriz A se puede expresar como en (Hartley & Zisserman, 2003):

$$A = R(\theta)R(-\varphi)DR(\varphi) \quad (33)$$

$$D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \quad (34)$$

La peculiaridad de la matriz de afinidad es el ángulo φ que relaciona a las escalas permitiendo tener deformaciones como de la Figura 23.

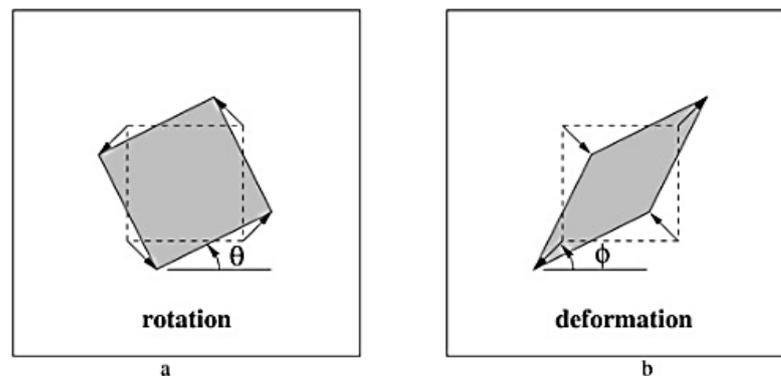


Figura 23. Distorsiones que surgen de la transformación afín plana. a) Rotación por $R(\theta)$. b) Deformación con el ángulo y rotación asociados $R(\varphi)$.

Fuente: (Hartley & Zisserman, 2003)

Para poder obtener esta transformación se necesitan tres pares de correspondencias entre puntos, teniendo así dos grados de libertad más que la transformación de similitud, (6 grados de libertad).

2.4.3 Transformación de Perspectiva

Finalmente esta transformación permite disponer de todos los parámetros asociados al movimiento de un objeto en tres dimensiones que se traducen mediante la proyección en un plano de dos dimensiones, los parámetros que contiene esta transformación expresan completamente la pose; las tres rotaciones y traslaciones posibles, la matriz asociada a esta transformación se mencionó anteriormente y es la matriz de homografía H que con su descomposición permite estimar el movimiento tridimensional, una transformación de perspectiva aplicada a los puntos pertenecientes al conjunto x se representa como (35):

$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (35)$$

De manera más resumida se puede representar como $x' = Hx$ donde se le aplica una transformación de perspectiva a los puntos de x para transformarlos en x' ; la cantidad mínima de pares de correspondencias necesarias para dar una respuesta exacta de esta transformación es cuatro de manera que se obtienen ocho grados de libertad con nueve parámetros de los cuales hay ocho parámetros linealmente independientes a tratarse como incógnitas de la matriz de transformación.

En la Figura 24 se puede apreciar las distorsiones bajo las transformaciones de similitud, afín y proyectiva, con sus diferencias.

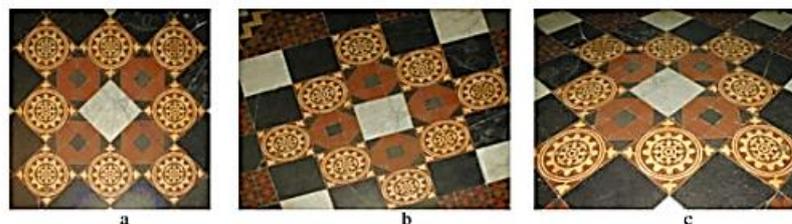


Figura 24. Distorsiones de proyección central debido a (a) Transformación de Similitud. (b) Transformación Afín. (c) Transformación de Perspectiva.

Fuente: (Hartley & Zisserman, 2003).

2.5 Control de un UAV

El uso de la teoría de control clásico en vehículos aéreos no tripulados es muy aceptable y utilizada en trabajos como (Bohdanov, 2012; Leos Monroy, 2014; Parra Muñoz, Feitosa, & Alves da Silva, 2013) donde por la naturaleza de este control se asumen aproximaciones lineales del sistema, los sistemas en cuestión tienen una cantidad de variables dependientes que deben ser controladas para obtener un comportamiento deseado. Las estrategias de control para una planta pueden ser de lazo abierto o cerrado (Vyškovský, 2014) Figura 25.

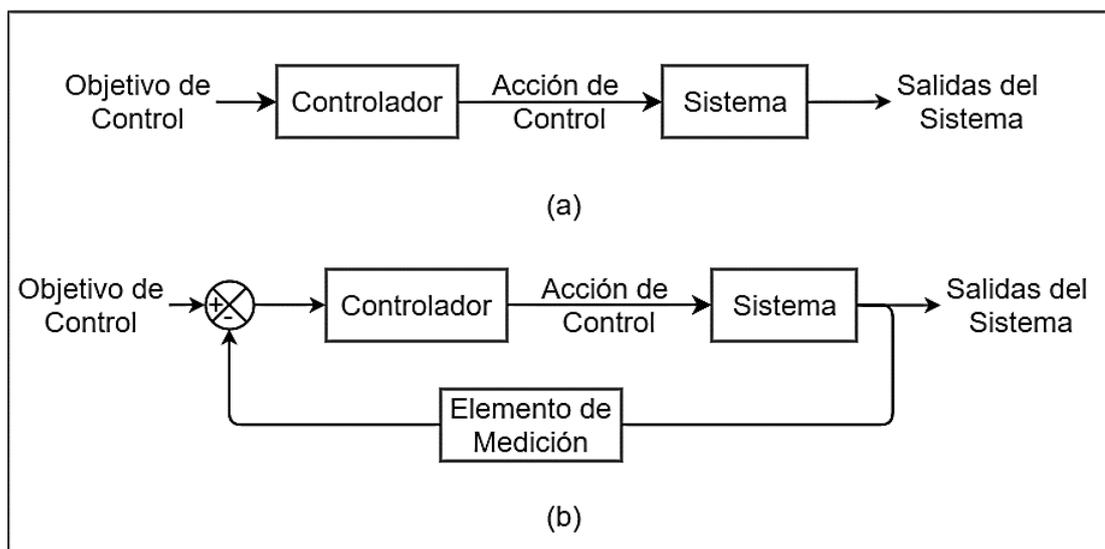


Figura 25. Esquemas de control. (a) En lazo abierto (b) En lazo cerrado.

Fuente: (Vyškovský, 2014)

El lazo abierto permite realizar el control del sistema siguiendo un objetivo (set point) sin influencia de la respuesta del sistema, por tanto no se toman en cuenta posibles errores o perturbaciones que influyan en la respuesta del mismo para manipular los valores de entrada al controlador. El control en lazo cerrado por otra parte permite realizar el seguimiento del set point teniendo una realimentación de la respuesta del sistema, de manera que se pueden corregir ciertas perturbaciones que incidan en el correcto comportamiento del mismo.

Uno de los controladores más utilizados para sistemas de control industrial es el PID Figura 26 que realiza acciones de control basado en la diferencia o desviación que

existe entre un valor medido y uno deseado, es decir, la entrada al controlador es el error de dicha diferencia y su salida la señal de control (Enrique & Fernández, 2016).

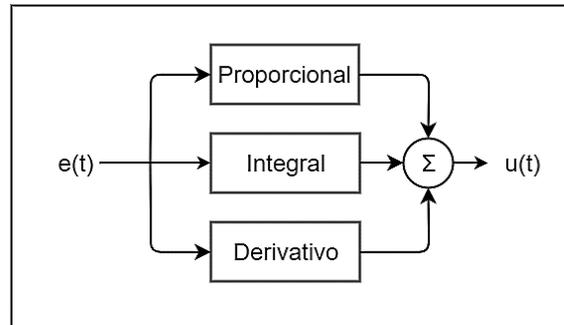


Figura 26. Diagrama de bloques del controlador PID en modo paralelo.

Fuente: (Enrique & Fernández, 2016)

En este controlador se usan tres tipos de acciones:

- **Acción Proporcional.** Se introduce en un sistema donde la acción de control es proporcional al error.

$$P = K_p e(t) \quad (36)$$

- **Acción Integral.** Esta acción es proporcional a la integral del error, opera en base a la memoria de los valores pasados acumulados de la señal brindando una señal de control que disminuye el error en estado estable.

$$I = K_i \int_0^t e(\tau) d\tau \quad (37)$$

- **Acción derivativa.** Es proporcional a la derivada del error y proyecta la tendencia de su crecimiento dando una señal de control que anticipa cambios grandes.

$$D = K_d \frac{d e(t)}{dt} \quad (38)$$

Finalmente este tipo de controlador permite influir en la dinámica del sistema así como en su estado estable para obtener el comportamiento deseado a lo largo del tiempo.

CAPITULO III

GENERALIDADES DEL SISTEMA

En este capítulo se detallan los componentes elegidos de hardware y software para el desarrollo del presente proyecto de investigación, así como las configuraciones de las plataformas que se han implementado para realizar las pruebas de funcionamiento del sistema, cabe mencionar que la propuesta de la investigación es el desarrollo de un sistema para ejecutar el aterrizaje automático de un Vehículo Aéreo no Tripulado en una plataforma móvil siendo ejecutado el control del vehículo aéreo desde una estación en tierra.

3.1 Hardware del Sistema

El presente proyecto ha sido desarrollado con el uso de tres componentes principales de hardware que son:

- Vehículo Aéreo no Tripulado.
- Plataforma móvil terrestre.
- Estación en tierra.

3.1.1 Vehículo aéreo no tripulado

El vehículo aéreo no tripulado seleccionado para este trabajo es el cuadricóptero Bebop 1 (Figura 27) de la empresa francesa Parrot el cual destaca por su bajo costo y por ser de código parcialmente abierto, pertenece a la escala VI, un micro UAV según (Custers, 2016; Hassanalain & Abdelkefi, 2017; Kendoul, 2012) es fácil de montar y con diez minutos de autonomía en el aire por el uso de una batería de litio, además cuenta con un ordenador muy potente incorporado y algoritmos de estabilidad de vuelo y video, algoritmos de seguridad de vuelo y regreso a casa; características que son esenciales para evitar sucesos emergentes, como la colisión, en cuyo caso se detienen las hélices inmediatamente. En un principio el drone ha sido pensado por la compañía para fines recreativos pero por sus características se puede utilizar en aplicaciones de simulación e investigación como en el presente proyecto, a continuación se presentaran algunos aspectos técnicos relevantes para este trabajo.



Figura 27. Bebop 1 de Parrot con las protecciones de espacio interior EPP
(«Parrot gallery», s. f.).

3.1.1.1 Estructura física

El drone tiene una estructura bastante robusta siendo reforzada con ABS de pluma, además de tener cascos EPP para los vuelos interiores dándole seguridad contra impactos, posee un ordenador dual core Parrot P7 que es 8 veces más potente que su antecesor el AR Drone. Tiene un procesador gráfico quad core y memoria flash interna de 8 Gb, montados en los soportes necesarios para su refrigeración y blindaje. Además de contar con un sistema GPS que sirve para varios aspectos de navegación, para lograr la estabilidad, usa la información de varios sensores entre los cuales se encuentran un acelerómetro, giroscopio y magnetómetro (todos de tres ejes), un sensor ultrasónico para la altura con ocho metros de alcance además de tener una cámara inferior que ayuda a la estabilización vertical y analiza la velocidad del drone mediante un procesamiento que incluye datos de la cámara inferior y el sensor ultrasónico.

La información del conjunto de sensores mencionados es analizada automáticamente por el drone para realizar acciones que mejoren la estabilidad y no interfieran en la maniobrabilidad.

3.1.1.2 Conectividad

La comunicación con el drone se realiza por WIFI mediante una red ad-hoc creada por el mismo dispositivo que tiene la posibilidad de manejar frecuencias MIMO de 2.4GHz y 5GHz configurables o autoseleccionados dependiendo de la interferencia presente en la red, la potencia de emisión de la señal es de hasta 21 dBm con un alcance de hasta 250 metros. Se debe destacar que el dispositivo fue pensado originalmente para ser controlado desde una Tablet o dispositivo celular de manera que se puede acceder a él mediante la aplicación oficial “Freeflight 3”, además cuenta con sistema

operativo Linux y su SDK (Kit de Desarrollo de Software) abierta para desarrolladores.

3.1.1.3 Capacidad y dimensiones

El drone es capaz de volar a velocidades de hasta 13 m/s de manera que tiene gran potencial en vuelos exteriores; sus dimensiones son de $28 \times 32 \times 3.6 \text{ cm}$ y 400 g sin protecciones interiores y de $33 \times 38 \times 3.6 \text{ cm}$ y 420 g con ellas Figura 28.

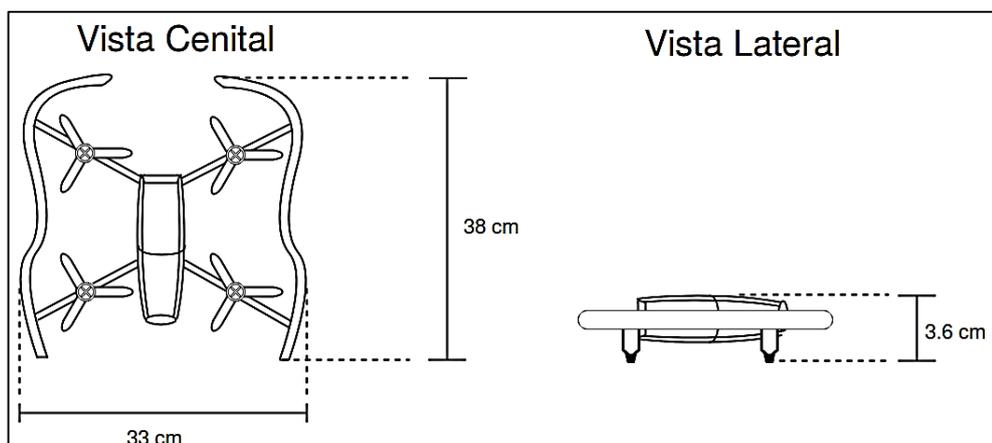


Figura 28. Dimensiones del drone Bebop 1 de Parrot en una vista cenital y lateral.

El tipo de control del drone está adaptado en configuración cruz por la disposición de su cámara delantera (descrito en el Capítulo II Tabla 2) como se aprecia en la Figura 29.

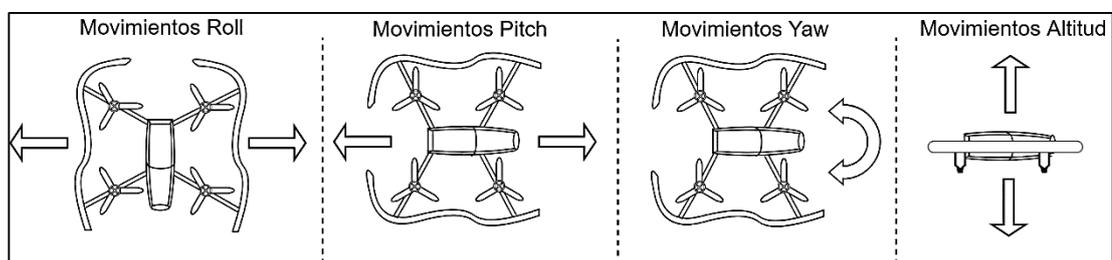


Figura 29. Movimientos del drone Bebop 1 de Parrot para cada acción de control.

3.1.1.4 Dispositivo de visión

Lo más importante para el presente proyecto es que cuenta con una cámara frontal de ojo de pez (180°) con apertura de 2.2", 6 elementos ópticos y sensor de 14 megapíxeles con una resolución de video de $1920 \times 1080p$ (30fps) con codificación H264 y foto de 3800×3188 , se incorporan algoritmos de estabilización exclusiva de tres ejes de imagen que son ejecutados en un microcontrolador incorporado destinado a esta función, permitiendo así obtener imágenes de alta calidad en un amplio rango y sin distorsiones a pesar de las turbulencias provocadas por el vuelo.

Se ofrece la característica de movimiento del punto de visión durante el video gracias al rango de 180° de la cámara, aunque hay que tomar en cuenta que a medida que la cámara virtual se ve enfocada a los extremos se pueden presentar distorsiones en la imagen al llegar a velocidades altas haciéndola borrosa. El ángulo de apertura que se presenta en video para la cámara del drone es de aproximadamente 50° en el eje vertical (ordenadas) y de 80° en el eje horizontal (abscisas) (Figura 30), así el rango de visión varía en función de la distancia de la cámara respecto al objetivo.

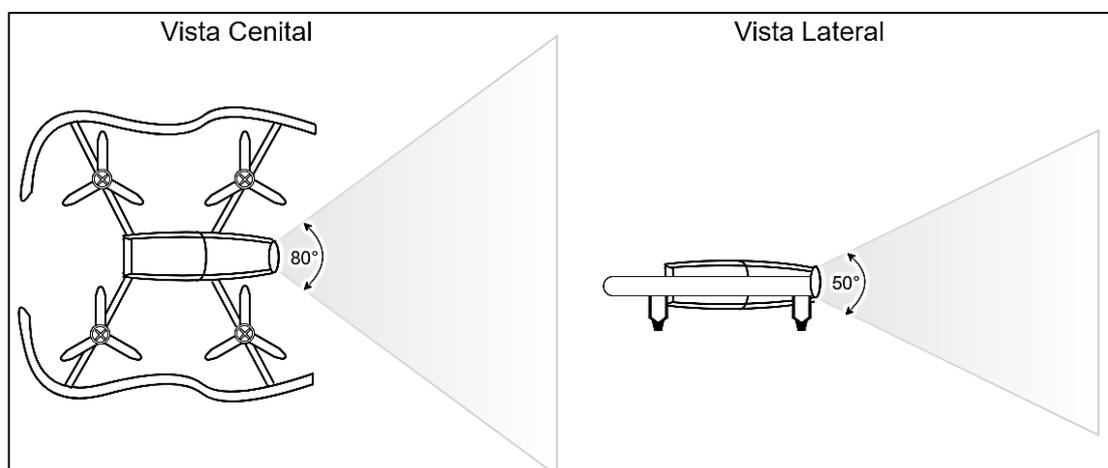


Figura 30. Ángulos de apertura de video para la cámara del Bebop 1 desde la perspectiva cenital y lateral.

Además el dron retorna un video completamente rectificado y sin distorsiones debido a los algoritmos que lleva incorporados que se encargan de suprimir la distorsión de perspectiva y mantienen un modelo proyectivo simple que conserva las propiedades de la escena captada traduciéndolas en imágenes 2D estables.

3.1.2 Plataforma terrestre

La plataforma terrestre elegida es un vehículo móvil de cuatro ruedas al cual se le han asignado el control de los motores de cada par de ruedas laterales a una sola señal, es decir el motor izquierdo controla las ruedas izquierdas y el motor derecho controla las ruedas derechas de modo que el vehículo permite movimientos en trayectorias rectilíneas y de rotación sobre su centro satisfactoriamente, las dimensiones del mismo son $30 \times 30 \text{ cm}$ con una altura de 20 cm como se puede apreciar en la Figura 31, ésta configuración permite una mayor estabilidad durante el movimiento ya que su centro de gravedad se encuentra situado en el centro del vehículo; adicionalmente sobre el mismo se ha incorporado una plataforma de madera plana de forma cuadrada y dimensiones $60 \times 60 \text{ cm}$ para que el drone tenga un lugar donde lograr estabilidad al aterrizar.

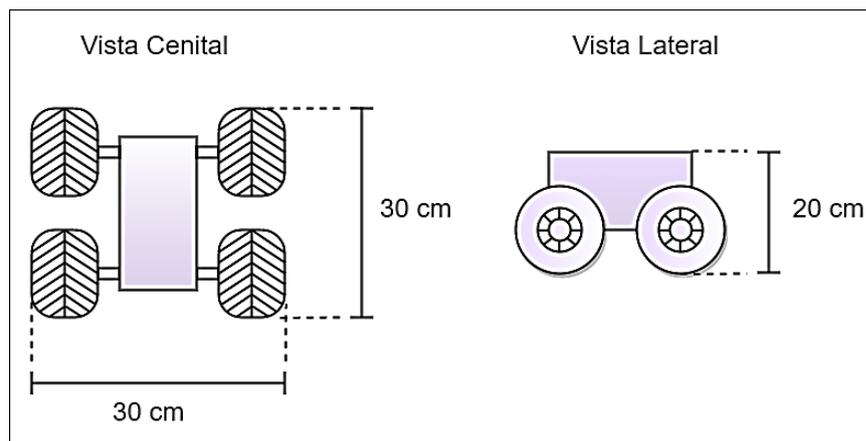


Figura 31. Dimensiones de la plataforma móvil terrestre.

Como se mencionó en la sección anterior, para detectar de manera confiable una colección de puntos de interés en una escena es necesario que la misma disponga de múltiples cambios de contraste, niveles de iluminación y diferencias en la imagen haciéndola de naturaleza no lisa ni uniforme. El uso de características locales brinda la opción de elegir aleatoriamente una imagen solo tomando en cuenta que tenga las características antes descritas, para este propósito se ha seleccionado una imagen de publicidad del drone como objetivo (target) como se puede observar en la Figura 32 la cual presenta múltiples cambios de contraste, esquinas, color y textura.



Figura 32. Imagen consigna elegida para el seguimiento (target).

La imagen de dimensiones $36 \times 30 \text{ cm}$ ha sido centrada en la plataforma de aterrizaje para realizar las pruebas correspondientes a la investigación. El montaje final se puede apreciar en la Figura 33.



Figura 33. Montaje final de la plataforma móvil terrestre.

Las dimensiones de la plataforma de aterrizaje con respecto al target permiten tener un rango de tolerancia correspondiente al aterrizaje en la plataforma móvil, la relación de aspecto respectivo se presenta en la Figura 34, además las dimensiones de dicha plataforma duplican las del drone el cual mide cerca de $28 \times 32 \text{ cm}$ sin las protecciones dándole un rango de tolerancia en el posicionamiento de aproximadamente la mitad de las dimensiones del drone, esto se puede apreciar en el esquema de la Figura 34 donde la permisibilidad en ambos ejes es de 15 cm y 18 cm en X, Y respectivamente con el drone totalmente dentro de la plataforma.

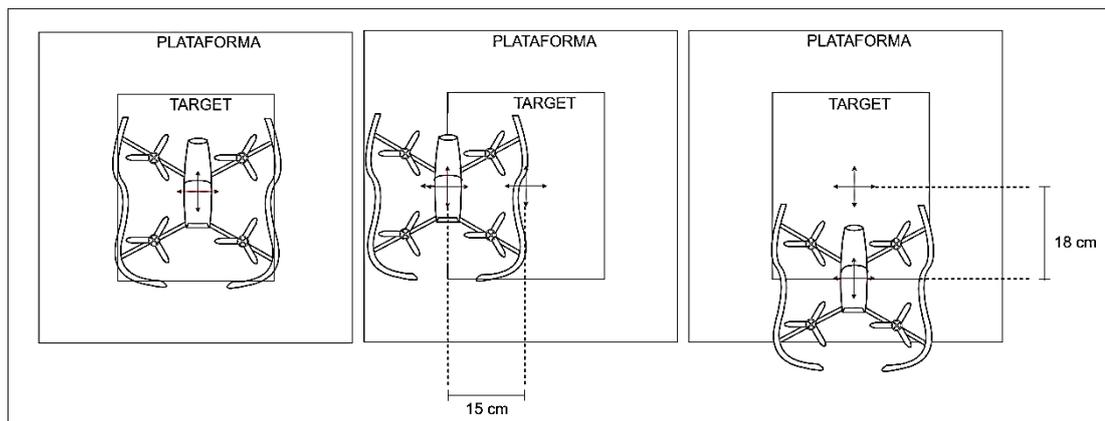


Figura 34. Relación de aspecto entre la plataforma de aterrizaje y el dron donde se aprecia la permisibilidad de posicionamiento.

3.1.3 Estación en tierra

La estación en tierra es un computador portátil capaz de realizar la conexión mediante WIFI con el dispositivo aéreo no tripulado y enviar órdenes mediante el uso de las herramientas instaladas para control del mismo, además cuenta con características suficientes para soportar las herramientas de software y algoritmos con respecto al costo computacional que estos requieren, las características principales del ordenador son:

- Fabricante: DELL Inc.
- Modelo: XPS L520X
- Procesador: Intel® Core™ i7-264M CPU @ 2.80GHz x 4
- Memoria RAM: 8 GB
- Tipo de sistema: Sistema operativo de 64 bits

El sistema operativo instalado es Ubuntu 14.04 LTS de 64 bits.

3.2 Software del sistema

Debido a los aspectos favorables de los elementos utilizados para el sistema, los procedimientos de software se desarrollan sobre el sistema operativo Ubuntu 14.04 LTS de 64 bits por ser uno de los más estables en las distribuciones de Linux, permite incorporar Bibliotecas de procesamiento de imagen como lo es OpenCV en la versión

instalada 3.1.0 más los paquetes adicionales de contribuciones disponibles en los repositorios, el cual proporciona una gran cantidad de herramientas de procesamiento de imágenes.



Figura 35. Íconos correspondientes de Ubuntu y OpenCV.

Fuente: («OpenCV homepage», s. f., «Ubuntu homepage», 2018)

3.2.1 Comunicación usando ROS

La comunicación entre la estación en tierra y el dron se realizó mediante el sistema operativo de robots ROS (Robot Operating System), que provee librerías y herramientas que permiten el desarrollo de software para robots mediante una arquitectura característica distribuida de grafos donde el procesamiento toma lugar en “nodos” y permite separar los procesos en pequeñas porciones mejorando el rendimiento de las aplicaciones empleando una topología de punto a punto donde los procesos se encuentran entre sí en tiempo de ejecución (Quigley et al., 2009).



Figura 36. Ícono del sistema operativo de Robots (ROS).

Fuente: («ROS (Robot Operating System)», s. f.)

El sistema de ROS es de código abierto y no tiene preferencias por algún lenguaje de programación, por tanto, se pueden generar códigos en C++, Python u otros sin problemas pues dispone de las herramientas necesarias para la compilación y ejecución de estos, una de sus filosofías es la reutilización de códigos de manera que se pueden adicionar fácilmente librerías de los repositorios (Serrano, 2011), ROS funciona mediante la ejecución de un conjunto de “nodos” que se comunican entre sí

compartiendo información con un modelo de mensajería publicador/subscriptor . Para poder comunicar los nodos se necesita de un master que proporciona información de conexión a los nodos permitiendo la transmisión de mensajes entre ellos, los mensajes se envían por canales de información llamados “tópicos” de manera que un nodo puede publicar o suscribirse a un tópico independientemente de si ya hay otros nodos haciéndolo (información unidireccional) pero si lo que se desea es una comunicación síncrona bidireccional existen los “servicios” que permiten la petición/respuesta de alguna información.

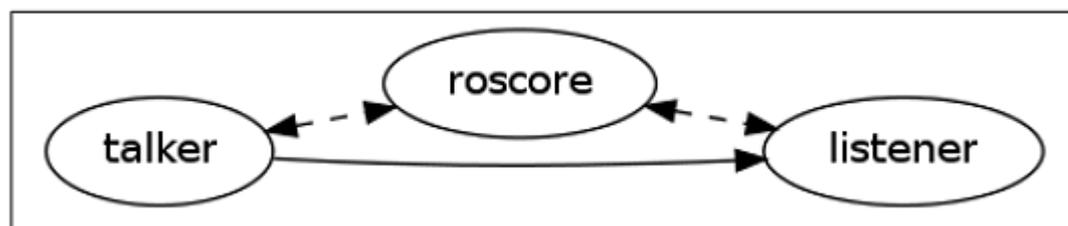


Figura 37. Esquema del funcionamiento básico de ROS con un nodo talker y uno listener comunicados gracias a un master (roscore).

Fuente: (Quigley, Gerkey, & Smart, 2015)

3.2.2 Uso del controlador de ROS para Bebop Drone

En los repositorios de ROS se hallan múltiples herramientas para el control de robots, entre los cuales están incluidos muchos UAV's que tienen abierta su SDK; “**bebop_autonomy**” es el controlador que ofrece ROS para los drones de Parrot (Bebop 1 y 2), el controlador está basado en la SDK oficial de Parrot ARDroneSDK3 (Mani Monajjemi (AutonomyLab, 2015), que permite usar las distintas funciones del dron e traduciéndolas en tópicos, servicios, mensajes y parámetros que pueden ser utilizados desde el sistema operativo. Este driver permite controlar las funciones principales como alzar vuelo, aterrizar, navegar, y más mediante los tópicos correspondientes (takeoff, land, cmd_vel, etc), en la Tabla 3 se pueden apreciar algunos de los tópicos utilizados con sus parámetros correspondientes.

Tabla 3
Tópicos fundamentales publicados por el driver bebop_autonomy con sus descripciones.

TÓPICO	TIPO DE MENSAJE	DESCRIPCIÓN
takeoff	std_msgs/Empty	Despegar verticalmente del suelo
land	std_msgs/Empty	Aterrizar de manera segura
reset	std_msgs/Empty	Detener los motores como emergencia
cmd_vel	geometry_msgs/Twist	Control de movimiento del dron
camera_control	geometry_msgs/Twist	Movimiento de la cámara virtual
image_raw	sensor_msgs/Image	Transmisión de video de la cámara frontal

Las órdenes publicadas en el tópico cmd_vel deben estar en el rango de $[-1 \dots 1]$ y son puestos en cero automáticamente al recibir algún comando en tópicos de mayor trascendencia como lo son takeoff, land o reset por razones de seguridad.

La calidad de video se ve limitada a una imagen de 640x368 o de 856x480 a 30Hz (dependiendo de la configuración de la cámara) la cual se puede captar en el tópico image_raw. Como se mencionó en la descripción del dron, image_raw se visualiza como una cámara virtual con la cual, mediante el driver, se puede variar la posición de captura teniendo adicionalmente un ángulo de apertura de entre unos 80° en horizontal y 50° en vertical.

Una cantidad mucho mayor de tópicos que los presentes en la Tabla 3 son publicados al arrancar el driver entre los cuales están tópicos de control para el usuario que permiten la navegación por GPS, secuencias de animación, toma de fotografías a bordo, grabación de video en alta resolución y otros de suscripción para lectura que tienen mensajes estándar de ROS como la Odometría y GPS, otros tópicos tienen tipos de mensajes creados para la aplicación y muestran la información referente al estado del dron. En versiones anteriores de Parrot como el AR-Drone esta información se conocía como Navdata con una transmisión constante y de alta frecuencia, característica que no tiene el Bebop, dado que la información de estados está limitada a publicarse a 5 Hz de manera selectiva y cuando los parámetros correspondientes al

estado se habilitan para lo mismo siendo una limitante de peso para el control completo de las funciones del drone.

Se puede arrancar el driver una vez que se haya conectado la computadora a la red creada por el drone y existen dos métodos de arranque: correr el driver como “nodo” y hacerlo como “nodelet”, la diferencia de estos es que los nodos son más fáciles de implementar y flexibles dando inicio a procesos de intercambio de datos TCP comunicados por direcciones provistas por un master, en cambio los nodelet pueden dar inicio a hilos comunicados mediante punteros a los datos manejando una estructura en ciertas ocasiones más rápida que con nodos.

Los archivos de arranque del driver se llaman “launch” o archivo de inicio el cual incluye todas las configuraciones necesarias para el arranque, así como las órdenes de ejecución del nodo correspondiente y los paquetes necesarios para ello. La ejecución del driver como nodo se realiza con el nodo `bebop_driver_node` del paquete `bebop_driver` y como nodelet con el nodo `bebop_nodelet` presente en el paquete `bebop_tools`.

Múltiples parámetros son configurados al arrancar el driver y le permiten al drone tener un correcto funcionamiento, por ejemplo parámetros para la asignación de una dirección IP, cuyo valor predeterminado es “192.168.42.1”, u otros que delimitan el ángulo máximo de inclinación para el movimiento en Yaw, Pitch, Roll, entre otros; estos parámetros pueden ser modificados acorde a las necesidades del usuario dando la posibilidad de personalización y mejor manejo del drone. Algunos de los parámetros más importantes en la configuración se indican a continuación en la Tabla 4.

Tabla 4
Parámetros de interés en la configuración del drone.

Parámetro	Descripción	Valor Actual
Parámetros de Pilotaje (prefijo <code>PilotingSettings...</code>)		
MaxAltitudeCurrent	Permite seleccionar la altura máxima en metros	30 m
MaxTiltCurrent	La inclinación máxima en grados para Pitch y Roll	20°
MaxDistanceValue	La máxima distancia hasta el punto de control	2000 m
MinAltitudeCurrent	La mínima altura para sostener vuelo	0.3 m
Parámetros de rapidez (prefijo <code>SpeedSettings...</code>)		
MaxVerticalSpeedCurrent	Máxima rapidez vertical	1 m/s
MaxRotationSpeedCurrent	Rapidez máxima de rotación en Yaw	100 °/s
MaxPitchRollRotationSpeed Current	Rapidez máxima de rotación en Pitch y Roll	200 °/s
Parámetros de Red (prefijo <code>NetworkSettings</code>)		
WifiSelectionBand	Banda con la que establece comunicación	Banda de 2.4 GHz
Parámetros de imagen (prefijo <code>PictureSettings</code>)		
VideoStabilizationMode	Selecciona los ejes de estabilización de video	Video plano en Pitch y Roll
VideoFramerate	Imágenes por segundo en el video	25 fotogramas por segundo

3.3 Extracción de imágenes

Para empezar con la etapa de procesamiento de imágenes se requiere acceder a la transmisión de video de la cámara frontal del drone, funcionalidad que se puede lograr a través del uso de ROS; con la utilización del controlador `bebop_autonomy` una vez arrancado en consola. Antes de dar inicio al controlador en un terminal se debe

direccionar los recursos del espacio de trabajo correspondiente al mismo para posteriormente lanzar la instrucción de inicio.

```
$ source ~/bebop_ws/devel/setup.bash
```

Para dar inicio al controlador se ha ejecutado la instrucción `roslaunch` como inicio en tipo nodo al ejecutar el nodo “bebop_driver_node” estableciendo una conexión de intercambio de información TCP.

```
$ roslaunch bebop_driver bebop_node.launch
```

3.3.1 Parámetros de imagen

Las imágenes captadas por el drone se publican en el tópico “**image_raw**” como un mensaje del tipo “`sensor_msgs/Image`” el cual es típico de ROS para codificar datos de imágenes, la resolución de las imágenes transmitidas es de 856×480 *px* a una tasa de 30 *Hz*, adicionalmente los datos de calibración se publican en el tópico `camera_info` donde se pueden evidenciar las matrices: matriz de calibración de la cámara (39), el vector de distorsión bajo el modelo de “Plumb Bob” que introduce coeficientes de distorsión tanto radiales como tangenciales debido a la naturaleza de la cámara de ojo de pez (40), la matriz de rectificación de imagen (41), y la matriz de proyección (42).

$$Camera_matrix = \begin{pmatrix} 537.292878 & 0 & 427.331854 \\ 0 & 527.000348 & 240.226888 \\ 0 & 0 & 1 \end{pmatrix} \quad (39)$$

$$Distortion_coefficients = \begin{pmatrix} 0.004974 \\ -0.000130 \\ -0.001212 \\ 0.002192 \\ 0 \end{pmatrix}' \quad (40)$$

$$rectification_matrix = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (41)$$

$$projection_matrix = \begin{pmatrix} 539.403503 & 0 & 429.275072 \\ 0 & 0 & 529.838562 \\ 238.941372 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (42)$$

3.3.2 Subscriptor de imágenes

El proceso de obtención de las imágenes publicadas en un tópico se llama subscriptor de imágenes, el nodo se debe subscribir al tópico que está publicando, en este caso el tópico correspondiente se llama `image_raw`.

La conexión entre el nodo que publica y el que escucha se puede comprobar con la herramienta `rqt_graph` de ROS que permite ver un gráfico dinámico del sistema Figura 38.

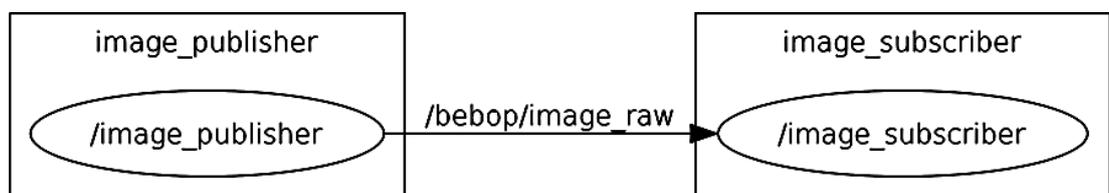


Figura 38. Gráfico dinámico del proceso de subscripción a imágenes en `rqt_graph`.

Para realizar el procesamiento de las imágenes obtenidas es necesaria una conversión de tipos de datos puesto que ROS publica las imágenes como tipo mensaje y OpenCV puede tratar imágenes en tipo matriz `cv::Mat`, para dicho propósito se utiliza la herramienta de conversión llamada `CvBridge` que realiza conversiones entre mensajes de ROS y los tipos de imágenes de OpenCV, como se evidencia en el siguiente diagrama (Figura 39):

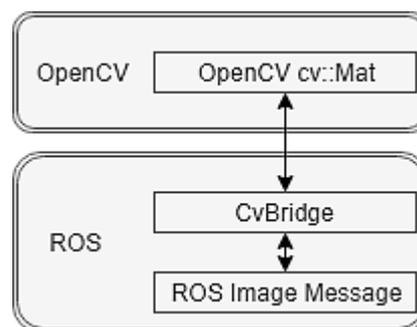


Figura 39. Diagrama de conversión de imágenes entre ROS y OpenCV.

Fuente: («ROS (Robot Operating System)», s. f.)

Para establecer la subscripción del nodo a un t3pico que publica im3genes se deben realizar los siguientes pasos:

- Creaci3n e inicializaci3n del nodo.
- Creaci3n e inicializaci3n del Subscriber tipo ImageTransport.

```
ros::init(argc, argv, "image_listener");
ros::NodeHandle nh;
image_transport::ImageTransport it(nh);
ros::Subscriber sub = nh.subscribe("bebop/image_raw", 1,
imageCallback);
```

En este proceso se especifican los par3metros correspondientes al t3pico del cual se van a recibir mensajes, el tama1o de la cola de mensajes y finalmente la llamada a la funci3n que realiza el intercambio de informaci3n.

Para llamar a la funci3n de intercambio se debe recurrir al comando “Spin()”, la funci3n encargada de realizar todo el proceso de obtenci3n y conversi3n de datos se presenta a continuaci3n:

```
void imageCallback(const sensor_msgs::ImageConstPtr& msg)
{
  /*Objeto cvBridge para conversi3n*/
  cv_bridge::CvImageConstPtr cv_ptr;

  try
  {
    /*Conversi3n entre tipos ROS-OpenCV */
    cv_ptr = cv_bridge::toCvShare(msg, "bgr8");
    /*La imagen se encuentra en cv_ptr->image*/
  }
  catch (cv_bridge::Exception& e)
  {
    ROS_ERROR("Could not convert from '%s' to 'bgr8'.", msg-
>encoding.c_str());
  }
}
```

Finalmente la imagen del tipo cv::Mat lista para su procesamiento se encuentra en “*cv_ptr->image*” donde se la extrae para su tratamiento en los procesos de caracterizaci3n de imagen del siguiente cap3tulo.

3.4 Control de movimiento del drone

Para controlar los movimientos del UAV y la orientación de su cámara se ha seleccionado los siguientes tópicos de la Tabla 3:

- `takeoff`.- Despegar el drone.
- `land`.- Aterrizar el drone.
- `reset`.- Detener los motores por emergencia.
- `cmd_vel`.- Control del movimiento del drone.
- `camera_control`.- Movimiento de la cámara virtual.

El controlador “`bebop_autonomy`” está suscrito a estos tópicos de manera que los mensajes publicados en ellos le sirven al controlador para enviar órdenes al drone, entonces para controlar sus movimientos se deben publicar los tipos de mensajes y parámetros correspondientes a cada uno de los tópicos anteriores.

Para publicar un mensaje en un tópico se debe seguir un procedimiento parecido a la suscripción pasando los parámetros correspondientes al tópico donde se publica y el tamaño de la cola de mensajes.

- Creación e inicialización del nodo.
- Creación e inicialización del Publicador.
- Publicación del mensaje

```

/*Inicializando el nodo Publicador*/
ros::init(argc, argv, "publisher");
ros::NodeHandle nh;
ros::Publisher pub = nh.advertise<msg_type> ("topic_name",
queue_size);

/*Publicando un mensaje*/
pub.publish(msg);

```

Los tópicos correspondientes al movimiento lineal del drone y la cámara virtual reciben mensajes de geometría “`geometry_msgs::Twist`” que tiene dos vectores de tres elementos cada uno, donde se pueden pasar parámetros lineales y angulares.

Los efectos de los parámetros correspondientes al tópico “cmd_vel” se presentan en la Tabla 5, y los del tópico “camera_control” en la Tabla 6.

Tabla 5
Efecto de los parámetros del mensaje tipo geometry_msgs::Twist en el tópico cmd_vel.

Parámetro	Valor	Acción
Linear.x	Positivo	Pitch adelante
	Negativo	Pitch atrás
Linear.y	Positivo	Roll izquierda
	Negativo	Roll derecha
Linear.z	Positivo	Ascender
	Negativo	Descender
Angular.z	Positivo	Yaw Rotar anti horario
	Negativo	Yaw Rotar horario

El tópico permite un rango de valores de los parámetros de [-1...1] en escala del máximo correspondiente a cada movimiento establecido en la configuración del controlador Tabla 4, para dejar estático al drone se debe publicar cero en cada uno de los parámetros.

Tabla 6
Efecto de los parámetros del mensaje tipo geometry_msgs::Twist en el tópico camera_control.

Parámetro	Valor	Acción
Angular.y	Positivo	Inclinación hacia abajo
	Negativo	Inclinación hacia arriba
Angular.z	Positivo	Panorámica hacia la derecha
	Negativo	Panorámica hacia la izquierda

Los valores de los parámetros en la Tabla 6 establecen la inclinación y giro panorámico en grados para mover la cámara en horizontal y vertical dependiendo del valor de parámetro.

3.4.1 Clase para manipulación del controlador “Bebop_autonomy”

Para facilitar el acceso a las funcionalidades expuestas se ha desarrollado una clase llamada “bebop_mov” para publicar mensajes en cada tópico cuando se lo requiera mediante la llamada a un método de la misma. El diagrama de clase se expone en la Figura 40.

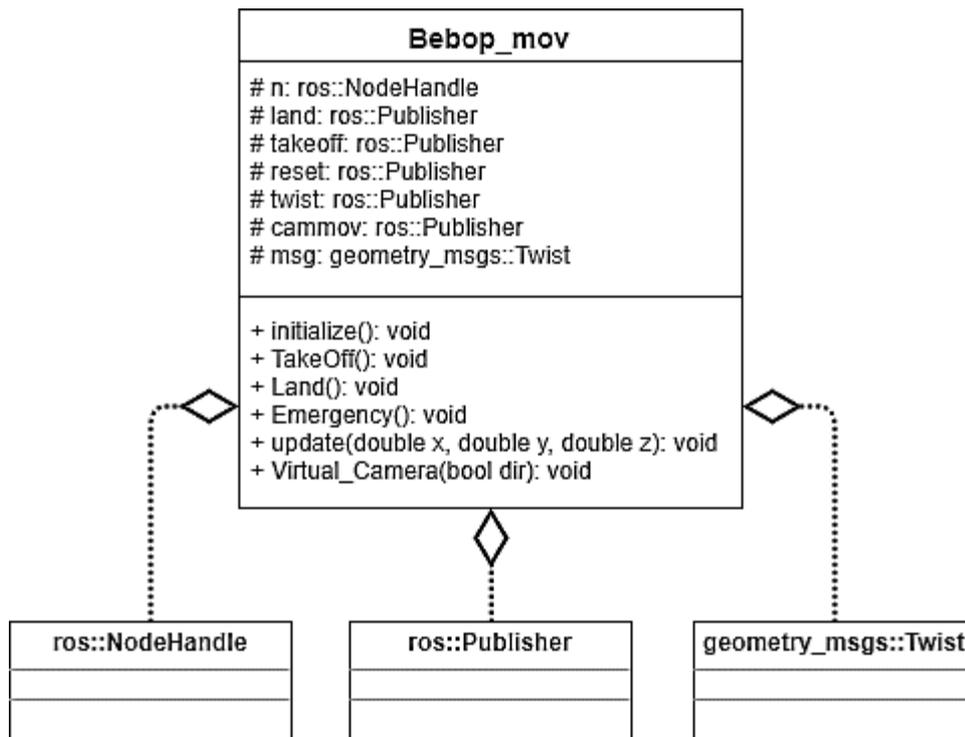


Figura 40. Diagrama UML de la clase Bebop_mov.

La clase “Bebop_mov” ha sido desarrollada de tal manera que permita realizar las acciones correspondientes a la Tabla 7 mediante la llamada a un método de la misma

Tabla 7
Función de los métodos de la clase *Bebop_mov*.

Método de la Clase	Función que desempeña
Initialize()	Inicializar los objetos “Publisher” para utilizarlos en los demás métodos.
TakeOff()	Publica mensajes en el tópico “takeoff” para despegue del drone
Land()	Publica mensajes en el tópico “land” para el aterrizaje del drone
Emergency	Publica mensajes en el tópico “reset” para el paro de emergencia del drone
Update()	Actualiza los valores de parámetros del movimiento lineal en (x, y, z) del tópico “cmd_vel”
Virtual_Camera()	Publica mensajes en el tópico “camera_control” para inclinar la cámara hacia abajo. (-70° máximo)

Adicionalmente se ha implementado un hilo de publicación constante en el tópico “cmd_vel” para mantener control del movimiento durante todo el proceso:

```
void drone_mov()
{
    ros::NodeHandle n;
    ros::Publisher twist;
    twist= n.advertise<geometry_msgs::Twist>("bebop/cmd_vel", 10);
    ros::Rate loop_rate(60);

    while (ros::ok())
    {
        twist.publish(msg);
        loop_rate.sleep();
    }
}
```

Con la implementación del hilo “drone_mov” basta con actualizar los valores de movimiento lineal en el método “Update()” para controlar el movimiento lineal del drone.

3.4.2 Comandos de detención del drone

Para realizar el aterrizaje del drone se puede acudir a dos de los comandos del controlador “Bebop_autonomy”:

- “Reset” que detiene los motores del drone haciendo que descienda en caída libre desde la posición en la cual se encuentra suspendido.
- “Land” este comando al igual que “Reset” y “TakeOff” pone a cero las órdenes de movimiento lineal en el drone, de manera que se mantiene suspendido en la posición en la cual se encuentra disminuyendo la velocidad de los motores hasta que aterriza.

Ambos comandos son de gran utilidad para detener al drone, el primero para hacerlo de inmediato, y el segundo para mantener la posición en la que se encuentra hasta su descenso.

Como se mencionó anteriormente el drone estima su velocidad y posición vertical con los datos provenientes del altímetro y las imágenes cenitales de la cámara inferior Figura 41. La cámara inferior toma una imagen cada 16 milisegundos para compararla con la anterior y determinar así la velocidad del drone, por esta razón los algoritmos internos del drone brinda estabilidad manteniendo su posición sobre la superficie que detecta de manera visual cuando no existen comandos de movimiento ejecutándose.

Esta condición permite que el comando “Land” no solo disminuya la velocidad de los motores sino también acompañe la posición visual que detecta la cámara inferior en la cual se lo ejecutó.



Figura 41. Localización del sensor de flujo óptico para estimación de la velocidad del drone.

Fuente: («Parrot gallery», s. f.)

3.5 Mensajes de estado del drone

Adicionalmente al arrancar el driver “Bebop_autonomy” el drone publica información acerca del estado de sus distintas funcionalidades, entre las cuales están la cámara, el GPS, la estructura física, conexión WIFI, etc. Anteriormente se vio como subscribirse al tópico que permite obtener la transmisión de video, otros de los tópicos publicados permiten obtener funcionalidades típicas de ROS como la Odometría, o específicas del drone como el estado de la batería y demás.

Los estados específicos del drone se han dividido en dos partes por el driver; en primer lugar los estados comunes donde se describe la estructura física del drone como su batería, estado del almacenamiento, señal wifi, sensores y otros; en los estados de “Ardrone3” se transmite los estados de video e imágenes, estado de pilotaje y los estados provenientes de la unidad de medición inercial del drone, GPS y otros.

Es de interés para este trabajo conocer el estado del drone durante el vuelo para lo cual se han implementado subscriptores para los tópicos que proporcionan información inercial y Odometría.

El proceso de subscripción es parecido al visto anteriormente con la única diferencia que en este caso no hace falta realizar el subscriptor mediante un objeto de “ImageTransport” sino con un objeto de ROS “NodeHandle” como se observa a continuación en un ejemplo de subscripción al tópico que proporciona la medición de altura del drone a partir de la zona de despegue:

```
ros::init(argc, argv, "listener_bebop_imu");  
ros::NodeHandle n;  
ros::Subscriber  
sub=n.subscribe("/bebop/states/ardrone3/PilotingState/AltitudeChanged",  
1000,AltitudeCallback);
```

Además se debe tomar en cuenta que para cada parámetro que se desee obtener, la suscripción se debe hacer al tópico que le corresponde y leer el tipo de dato que es proporcionado. Para el caso de la lectura del parámetro altura le corresponden las siguientes características:

- Parámetro: ~states/enable_pilotingstate_altitudechanged
- Tópico: states/ardrone3/PilotingState/AltitudeChanged
- Tipo de mensaje: bebop_msgs::Ardrone3PilotingStateAltitudeChanged

El tipo de mensaje de altura creado en ROS es un dato flotante acompañado de un encabezado de reconocimiento, de manera que la función de obtención del dato (Callback) para la medición de altura se implementa de la siguiente manera:

```
void AltitudeCallback(const
bebop_msgs::Ardrone3PilotingStateAltitudeChanged::ConstPtr& msg)
{
    Float 64 altura=msg->altitude;
}
```

De igual manera se han realizado subscriptores para los parámetros de la Tabla 8:

Tabla 8
Parámetros y características para las suscripciones implementadas.

Parámetro	Tópico	Tipo de Mensaje
Estado de navegación		
Odometría	odom	nav_msgs::Odometry
Estados de Pilotaje (prefijo Ardrone3PilotingState)		
Cambio de altitud	(prefijo)AltitudeChanged	bebop_msgs:: (prefijo)AltitudeChanged
Cambio de postura	(prefijo)AttitudeChanged	bebop_msgs:: (prefijo)AttitudeChanged
Cambio de velocidad	(prefijo)SpeedChanged	bebop_msgs:: (prefijo)SpeedChanged

- Odometría. Retorna la pose del drone basado en la fusión de los datos inerciales y visuales. (la medición de Odometría proporcionada por el drone no es exacta y presenta inexactitudes)
- Altura. Retorna la medición de altitud con respecto al piso
- Rapidez. Retorna la velocidad lineal en m/s para los ejes x, y, z
- Postura. Retorna la inclinación en Yaw, Pitch y Roll del drone

El uso de esta información es de gran ayuda para controlar al drone, no obstante, el drone solo la transmite a una tasa aproximada de 5Hz, de manera que la utilización de los mensajes de estado del drone sirven en su mayoría para obtener información ilustrativa ya que el control basado en mediciones inerciales requiere una rapidez de retroalimentación de los estados mucho mayor para evitar percances y aumentar la rapidez de respuesta del drone.

CAPITULO IV

DETECCIÓN Y SEGUIMIENTO VISUAL DEL TARGET

En este capítulo se presentan los procedimientos correspondientes a la fase de captura, detección y seguimiento de puntos de interés en la escena empezando por la comunicación con el dron para seguir por el tratamiento de las imágenes con una evaluación de los algoritmos que presenten mejor rendimiento para la aplicación y finalmente se presenta la propuesta de correspondencia entre fotogramas consecutivos para mayor robustez en la detección.

4.1 Detección, descripción y emparejamiento de puntos de interés del target

Para identificar al objetivo (target) del resto de la escena es necesario recurrir a los algoritmos de detección de características locales en la imagen, para ello se han implementado tres de los más representativos métodos; el algoritmo SIFT un extractor de características muy ampliamente utilizado en la literatura por su eficiencia, SURF que mejora al algoritmo SIFT en rapidez además de una precisión equiparable y ORB que es uno de los métodos más novedosos y permite optimizar el procesamiento haciendo de este más rápido y con buen desempeño como se mencionó en el Capítulo II, los tres algoritmos permiten la detección a múltiples escalas haciéndolos invariantes a la distancia en la cual se toman las imágenes.

Un criterio de eficiencia de los métodos de caracterización de imágenes es la cantidad de puntos que es capaz de detectar en el tiempo (Ilbay et al., 2013). Para diferenciar el target de una escena con múltiples cambios de contraste se necesita un detector capaz de encontrar gran cantidad de peculiaridades suficientemente representativas para su correspondencia y repetitividad.

4.1.1 Evaluación de algoritmos de detección de puntos de interés del target

En primer lugar se debe evaluar la cantidad de puntos máximos y tiempo de detección por cada método, para dicho fin se han aplicado los algoritmos de detección en una secuencia de cuatro imágenes tomadas del target (Figura 32) y una que difiere del mismo dando como resultado los datos de la Tabla 9, las imágenes tomadas tienen las siguientes características:

- (1) Imagen del target tomada a 80 cm.
- (2) Imagen del target tomada a 50 cm
- (3) Imagen inclinada del target tomada a 50 cm
- (4) Imagen del target tomada a 1.2 m y rotada
- (5) Imagen diferente al target

Tabla 9
Detección máxima de puntos de interés para los algoritmos SIFT, SURF y ORB.

Método	Métrica	Imagen 1	Imagen 2	Imagen 3	Imagen 4	Imagen 5
SIFT	N° puntos	3292	7693	5697	327	3241
SURF		16993	195781	45513	890	23382
ORB		12466	33301	22435	1185	9404
SIFT	Tiempo [s]	0.0476	3.7764	1.0955	0.0408	0.5157
SURF		1.1771	14.2567	3.4617	0.0460	1.7046
ORB		0.0402	0.3048	0.1454	0.0071	0.0839

Los algoritmos utilizados para el procesamiento SIFT, SURF y ORB han sido implementados mediante las funciones de dichos métodos que están disponibles en OpenCV dentro del marco de características 2D y permiten la creación de objetos del tipo SIFT, SURF y ORB con sus respectivos parámetros. Uno de estos parámetros es la cantidad de puntos de interés a retornar donde se obtienen los “n” puntos de interés más significativos detectados en la imagen, en SURF no se pasa el parámetro directo del número de puntos a retornar, en contraste, se pasa un parámetro del umbral de la matriz Hessiana para la detección de puntos dando los mejores resultados con este umbral.

Adicionalmente se pueden cambiar los parámetros de la pirámide de detección a múltiple escala o las capas de detección en cada caso; los niveles de detección han sido tomados por defecto así como los umbrales típicos de cada método, cambiando solo la cantidad de puntos a ser detectados dando un máximo para cada imagen.

La eficiencia que denotan las pruebas realizadas en la secuencia de imágenes de la Tabla 9 muestra el desempeño superior del algoritmo ORB con respecto a los otros dos por el tiempo de detección, la cantidad de puntos de interés con respecto al tiempo

de procesamiento se puede observar en la Figura 42 como un índice normalizado donde se evidencia la superioridad en este aspecto de ORB.

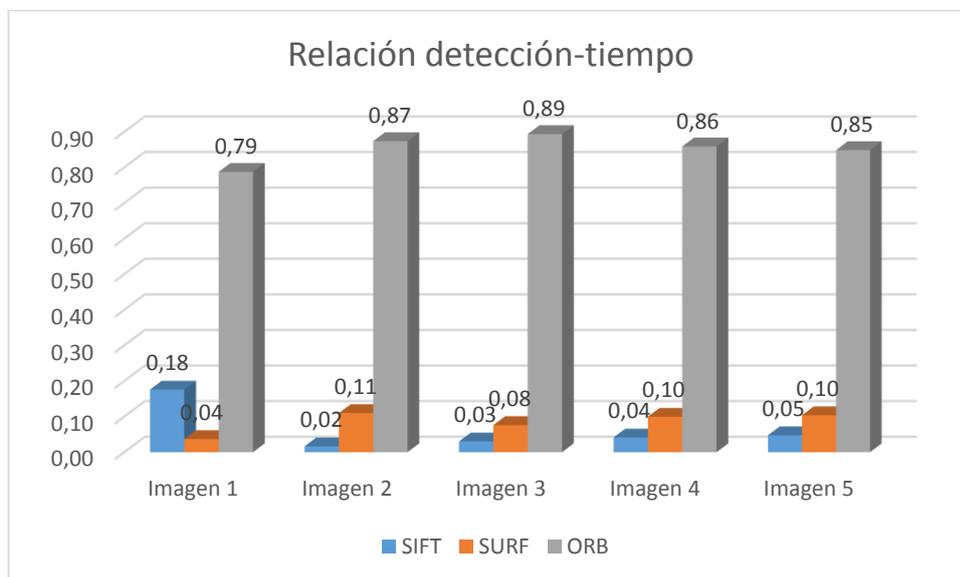


Figura 42. Relación detección-tiempo de los algoritmos SIFT, SURF y ORB para la secuencia de imágenes de la Tabla 9.

En la Figura 43 se puede apreciar un ejemplo de la detección y localización de 25 puntos de interés que incluyen orientación en sus gráficas.

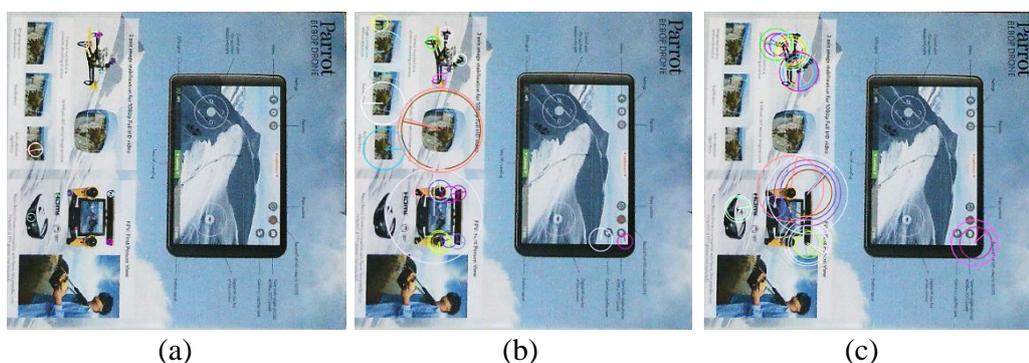


Figura 43. Detección de 25 puntos de interés en la imagen del target. (a) Algoritmo SIFT. (b) Algoritmo SURF. (c) Algoritmo ORB.

La cantidad de puntos de interés detectados en el target con respecto al tiempo que le toma hacerlo al algoritmo es importante para su discriminación, pero no es suficiente, estos puntos deben tener una descripción peculiar comparándolos con el resto de la imagen para poder discriminarlos en diferentes escenas con un buen índice de precisión y repetitividad.

4.1.2 Discriminación del target en movimiento

Es de interés determinar una buena detección en diferentes escenarios para poder evaluar la propiedad de discriminación que tienen los algoritmos al ser aplicados a una secuencia de imágenes con cambios en el punto de visualización. Este es el caso de un objetivo en movimiento donde interfieren los puntos de interés correspondientes a la escena fuera del target y los puntos del target. El target va cambiando su estructura geométrica en la imagen por los cambios de visualización de la cámara.

El proceso de discriminación de un target en una escena amplia se realiza registrando las características locales del target para luego compararlas con las características locales extraídas de cada fotograma consecutivo en la escena.

4.1.2.1 Correspondencias entre imágenes

El proceso de emparejamiento se realiza para poder determinar cuáles puntos de la imagen objetivo (el target) se visualizan correctamente en la imagen de evaluación actual, este proceso se ejecuta con el algoritmo de “fuerza bruta” de OpenCV debido a su comportamiento determinista que toma un descriptor de un primer conjunto (la imagen consigna) y lo compara con todos los descriptores del segundo conjunto calculando la menor distancia entre ellos como se mencionó en el Capítulo 2, el algoritmo de fuerza bruta puede ser usado en una versión del único punto como mejor candidato, y otra como los candidatos más cercanos al descriptor consigna; esta variación se refiere a los vecinos más cercanos en sus distancias euclidianas al descriptor consigna.

Para comparar los puntos de interés con el algoritmo de fuerza bruta se debe tomar en cuenta el parámetro de decisión para emparejar en los métodos correspondientes (SURF, SIFT y ORB) como una medición de las distancias de los descriptores. En OpenCV se pueden elegir las normas en función del tipo de descriptor utilizado (Kaehler & Bradski, 2016); para SURF y SIFT existen dos medidas de las cuales se puede seleccionar entre la norma NORM_L1 y NORM_L2, la diferencia de estas normas es que la primera usa una medida de la distancia de “Manhattan” donde se intenta minimizar la suma de las diferencias absolutas entre dos vectores descriptores y en la segunda norma se minimiza la suma del cuadrado de las diferencias, la norma

L2 es más precisa que L1 por tanto será utilizada para los algoritmos SIFT y SURF. Las distancias según las normas L1 y L2 para un conjunto de descriptores (\vec{a}, \vec{b}) se define como (43) y (44) respectivamente:

$$dist(\vec{a}, \vec{b}) = \sum_i abs(a_i - b_i) \quad (43)$$

$$dist(\vec{a}, \vec{b}) = \sqrt{\sum_i (a_i - b_i)^2} \quad (44)$$

Para ORB la métrica no se puede obtener como distancias euclidianas pues el descriptor que genera está basado en cadenas binarias, por tanto, se utiliza una medida de la distancia de Hamming como NORM_HAMMING y se define por (45):

$$dist(\vec{a}, \vec{b}) = \sum_i (a_i == b_i) ? 1 : 0 \quad (45)$$

Cabe mencionar que entre más puntos se detectan en una escena es más posible encontrar coincidencias satisfactorias con el target, no obstante esto puede ocasionar problemas de superposición de puntos haciendo que se generen falsas correspondencias que cumplen con el umbral de distancia entre descriptores necesario para considerarlas acertadas.

Para obtener los resultados deseados con el menor número de fallas se han aplicado dos métodos de eliminación de falsas correspondencias.

- Distancias entre vecinos cercanos.
- RANSAC

Para aplicar estos conceptos se ha utilizado el emparejador de fuerza bruta de OpenCV con conceptos de los k vecinos más cercanos, siendo para estos fines $k = 2$ para obtener las dos coincidencias mejores con respecto al descriptor consigna.

La distancia entre los vecinos cercanos sirve para eliminar las correspondencias que no pertenecen entre los frames pero no pueden ser descartados por un umbral global en la imagen, el concepto fue introducido por (David G Lowe, 2004) donde se comparan las distancias entre el vecino más cercano y el segundo vecino más cercano

y analiza las funciones de densidad de probabilidad entre las correspondencias correctas e incorrectas, se utiliza este método para eliminar las correspondencias que no pertenecen al mismo punto pero tienen características parecidas.

Para considerar un par de puntos como una correspondencia correcta se debe mantener la relación de distancia menor a un valor predefinido. El valor se toma como 0.8 en las pruebas de Lowe, con esto se eliminan el 90% de coincidencias falsas y se descartan menos del 5% de coincidencias correctas, por tanto es el mismo valor que será utilizado en este trabajo.

Posteriormente la utilización de la técnica iterativa llamada RANSAC sirve para eliminar los puntos que no se ajusten al modelo obtenido por el conjunto de pares de correspondencias correctas, descartando los valores atípicos dentro del mismo y dejando así solo las correspondencias más fiables y que se ajustan al modelo. En el diagrama de la Figura 44 se puede apreciar un esquema del proceso de emparejamiento de un target con una secuencia de imágenes que incluyen al target.

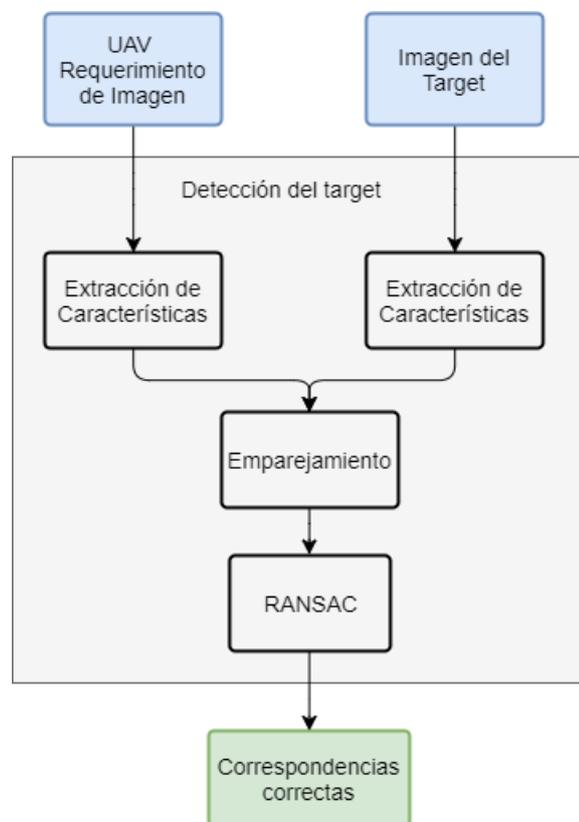


Figura 44. Diagrama de bloques de la detección del target en una secuencia de fotogramas.

4.1.2.2 Evaluación del desempeño

Los algoritmos de extracción de características pueden ser evaluados bajo criterios de precisión y eficacia al determinar cuántas correspondencias correctas se alcanzan con respecto a varias premisas.

Los criterios de evaluación presentados en (Işık & Özkan, 2014; Mikolajczyk & Schmid, 2005) determinan la eficacia del algoritmo con el concepto de recall y precisión.

Recall es la medida del cociente entre el número de correspondencias correctas detectadas y el número total de correspondencias (46), y la precisión se mide como el cociente entre el número de correspondencias correctas y el número total de posibles coincidencias en la imagen (47).

$$recall = \frac{\# \text{ correspondencias correctas}}{\# \text{ total correspondencias}} \quad (46)$$

$$precisión = \frac{\# \text{ correspondencias correctas}}{\# \text{ correspondencias posible}} \quad (47)$$

Una vez aplicados los algoritmos anteriores, se pueden extraer los parámetros correspondientes para evaluar el desempeño de cada método de extracción de características, para lo cual se han realizado un conjunto de pruebas en secuencias de video para discriminar la detección más confiable. Las secuencias de video son tomas del target en movimiento variando así la geometría del mismo, los movimientos cumplen con cambios de perspectiva, rotación, escalamiento hasta 1.8 m y en el tercer video se pierde el target por un tiempo determinado.

En la Tabla 10 se pueden apreciar los valores medios de las medidas de eficacia de cada método de extracción de características en cuatro secuencias de video tomadas con presencia del target que presentan cambios significativos en la geometría del mismo, así como la captura a distintas distancias y niveles de iluminación.

Tabla 10
Medidas de eficiencia de los algoritmos SIFT, SURF y ORB en secuencias de video.

Método	Métrica	Video 1	Video 2	Video 3	Video 4
SIFT	Recall	0.2533	0.3018	0.2288	0.1401
SURF		0.3603	0.4224	0.3151	0.1704
ORB		0.2979	0.4132	0.3207	0.2068
SIFT	Precisión	0.5354	0.5796	0.4885	0.3601
SURF		0.6230	0.6498	0.5399	0.3632
ORB		0.6124	0.6795	0.5990	0.4864
SIFT	Tiempo	0.2202	0.2247	0.2004	0.2058
SURF		0.3170	0.1449	0.1359	0.1258
ORB		0.0139	0.0137	0.0136	0.0425

Para ejemplificar estos resultados las figuras más adelante representan las métricas de Recall, precisión y tiempo de procesamiento para cada fotograma consecutivo comparado con el target en la secuencia de video 2.

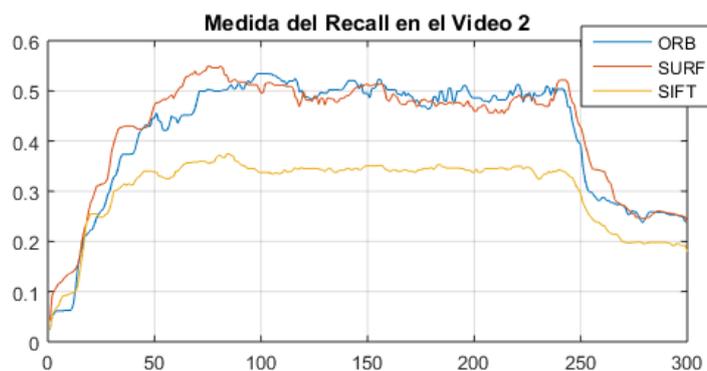


Figura 45. Medidas de recall para cada fotograma consecutivo en la secuencia de video 2 con los algoritmos SIFT, SURF y ORB.

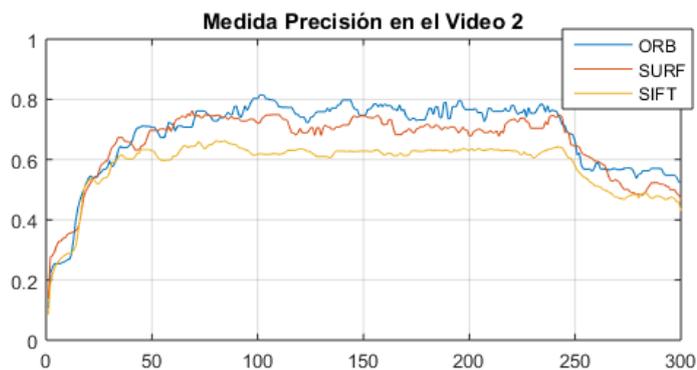


Figura 46. Medidas de precisión para cada fotograma consecutivo en la secuencia de video 2 con los algoritmos SIFT, SURF y ORB.

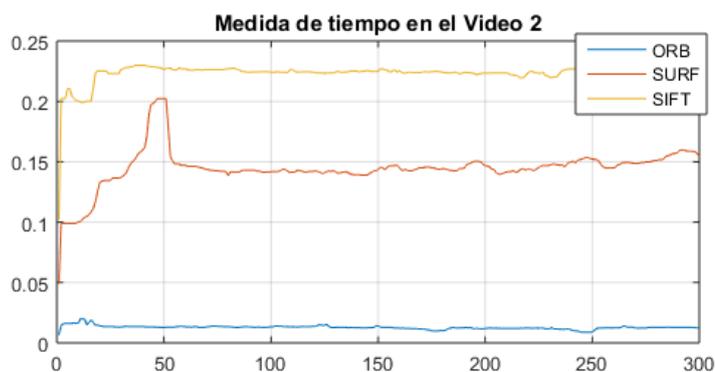


Figura 47. Medidas de tiempo de procesamiento para cada fotograma consecutivo en la secuencia de video 2 con SIFT, SURF y ORB.

Como resultado de aplicar los algoritmos de extracción de características y los respectivos métodos de eliminación de falsas correspondencias se puede observar en la Tabla 10 que los algoritmos SURF y ORB son equiparables en la mayoría de los casos, no obstante el tiempo de ejecución de ORB supera por mucho a los otros dos algoritmos haciéndolo más eficiente y preciso, de manera que este método será utilizado durante el resto del trabajo.

4.2 Seguimiento visual y posición relativa del target

Para realizar la detección satisfactoria del target en una imagen se necesita superar un umbral de características emparejadas satisfactoriamente, el cual luego de ser superado se puede asumir que se ha encontrado el objetivo dentro de la imagen.

4.2.1 Umbral de decisión

El umbral de decisión depende de la cantidad de puntos que se ven emparejados satisfactoriamente bajo diferentes condiciones de visualización, para este fin se han realizado un conjunto de pruebas en espacio interior y exterior donde se emparejan las características del ambiente para luego posicionar el target delante de la cámara, se tomaron 4 secuencias de video en un espacio interior Tabla 11 y 3 en espacio exterior Tabla 12 para diferentes alturas con respecto al target obteniendo los resultados a continuación.

Tabla 11

Correspondencias con el target en espacio interior.

		Video 1	Video 2	Video 3	Video 4
Altura de Detección	[m]	1.0	1.2	1.4	1.6
Zona no Detección	N° puntos	17.08	11.66	12.75	19.59
Zona de Detección		158.98	178.198	127.12	130.125
Valor medio		88.03	94.929	69.935	74.8575

Tabla 12

Correspondencias con el target en espacio exterior.

		Video 5	Video 6	Video 7
Altura de Detección	[m]	1.2	1.4	1.6
Zona no Detección	N° puntos	42.114	26.19	29.62
Zona de Detección		149.59	129.6	119.94
Valor medio		95.852	77.895	74.78

Dados los resultados de la Tabla 11 y Tabla 12 tanto en el caso de espacio interior como exterior el umbral tiene un salto del cual el valor medio oscila alrededor de “**70 puntos**”, por lo tanto ese es el umbral que será tomado para el emparejamiento. En la

Figura 48 y Figura 49 se puede apreciar el número de correspondencias en las secuencias de espacio interior y exterior respectivamente con la referencia de 70 puntos.

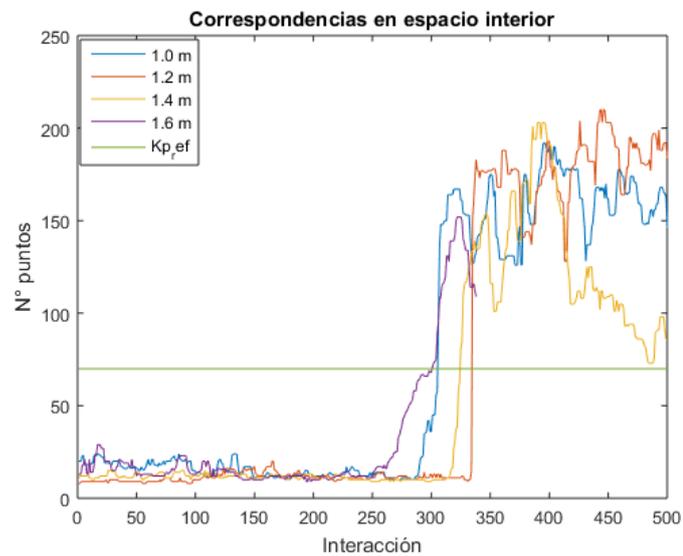


Figura 48. Correspondencias obtenidas en espacio interior con respecto a la referencia de 70 puntos.

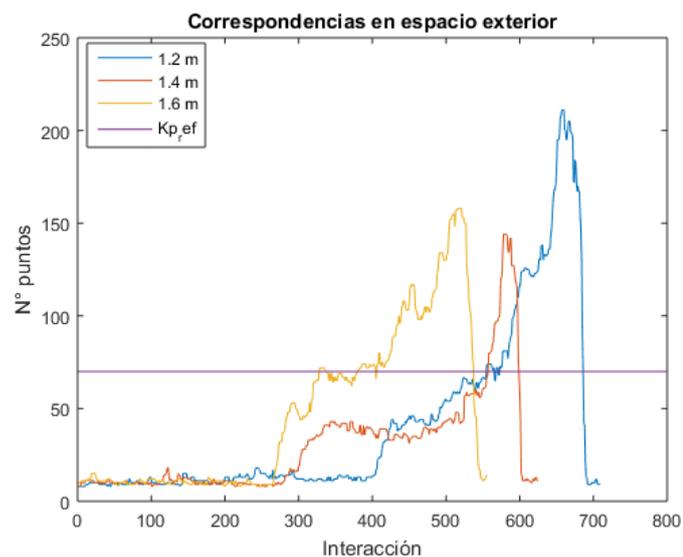


Figura 49. Correspondencias obtenidas en espacio exterior con respecto a la referencia de 70 puntos.

4.2.2 Encapsulamiento del target

Las características de los puntos emparejados satisfactoriamente entre la imagen del target y la imagen donde se lo busca representan las deformaciones y cambios geométricos que las relacionan. Si se define a la imagen consigna como I_0 y la imagen donde se lo va a buscar como I_1 y a los respectivos puntos emparejados satisfactoriamente como P_0, P_1 entonces existe una transformación geométrica que puede convertir a los puntos P_0 en P_1 .

Una manera de estimar los cambios geométricos sufridos por I_0 que se visualizan en I_1 es estimar la transformación geométrica de P_0 a P_1 . La transformada de Homografía es la que más fidelidad presenta a cualquier deformación en el espacio euclidiano por lo que se realiza la estimación de dicha transformada entre dos conjuntos de puntos.

OpenCV permite realizar cada una de las transformaciones geométricas del Capítulo 2 (sección 2.4), se puede encontrar la transformación de perspectiva con la matriz de homografía entre dos conjuntos de puntos gracias a la función “findHomography”; la función retorna una matriz de 3×3 con los nueve elementos típicos de una homografía y sus valores que representan los cambios geométricos sufridos de P_0 a P_1 . Adicionalmente la función permite descartar los puntos incorrectos mediante RANSAC y extraer una máscara con los puntos restantes (correctos).

```
if(matched1.size() >= 4) {
    homography =
    findHomography(Points(matched1), Points(matched2), CV_RANSAC,
    ransac_thresh, inlier_mask);
}
```

Para representar visualmente la detección del target en I_1 se realiza una transformación de perspectiva con la matriz de homografía obtenida trasladando el recuadro de las dimensiones del target a un recuadro que lo encapsule en el fotograma correspondiente de la siguiente manera:

```
perspectiveTransform(object_bb, new_bb, homography);
```

Donde “object_bb” es un conjunto de cuatro puntos que forman al cuadrado que encapsula al target en la imagen I_0 y new_bb es el nuevo recuadro que representa la transformación de perspectiva aplicada con la matriz de homografía Figura 50.

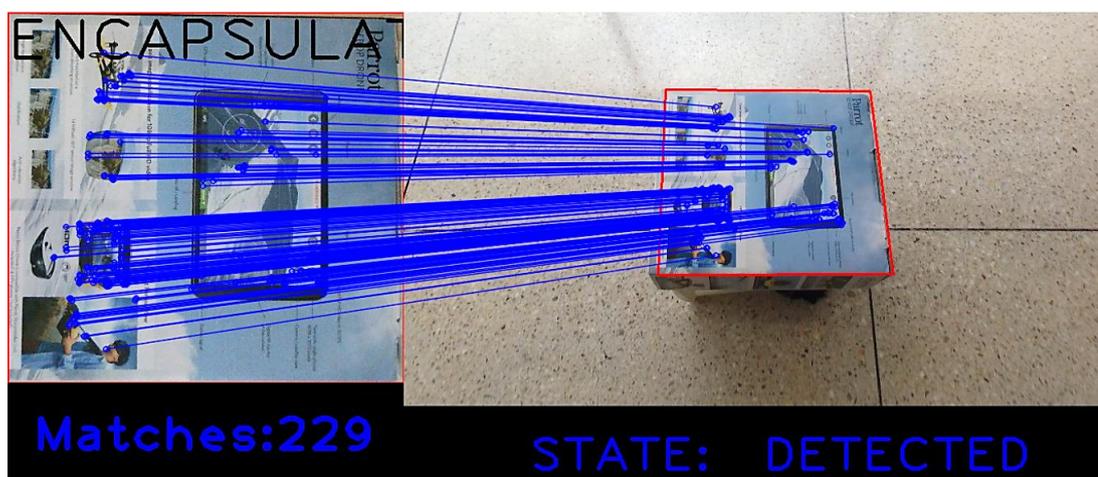


Figura 50. Encapsulamiento del target con la transformación de perspectiva.

4.2.3 Problemas en la detección

Cuando una cámara se encuentra en movimiento, como es el caso de los UAV que realizan tomas mientras siguen trayectorias, las maniobras agresivas y movimientos de la cámara con respecto a la posición de un target hacen del reconocimiento de un objeto mediante puntos de interés un proceso dependiente de los ángulos de visión, inclinación, cambios de luz, calidad de imagen y otros.

Estos factores influyen en la correcta discriminación de características locales en la imagen produciendo variaciones considerables al momento de emparejar dichas características, esto hace que el seguimiento de un target basado en el emparejamiento de características locales pierda precisión dado que para poder identificar un target en una escena se requiere pasar del umbral de puntos característicos detectados satisfactoriamente.

Efectos de pérdida de calidad en la imagen como la captura a velocidades grandes y escalamiento muy extenso producen pérdida de puntos de interés en la detección del target introduciendo la incertidumbre de si está o no el target a la vista. Algunos ejemplos se pueden visualizar en la Figura 51 donde se comparan las detecciones

incorrectas observando que con los efectos mencionados no se diferencia entre los puntos que están correctamente emparejados y los que no.

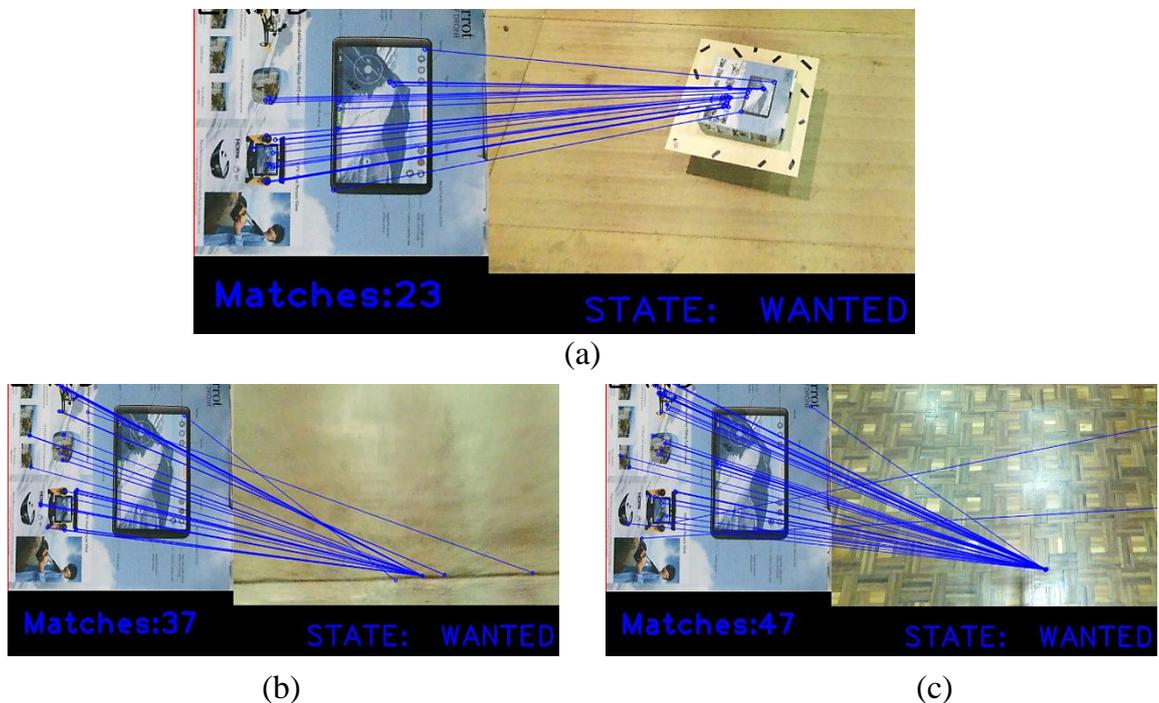


Figura 51. Errores en la detección por: (a) escala, (b) baja calidad en imagen y (c) cambios de textura e iluminación.

Para solventar este problema se ha optado por realizar la detección inicial bajo el concepto del target como fotograma consigna con el umbral de 70 puntos para posteriormente aplicar un método similar al presentado en (Aguilar & Angulo Bahón, 2013) donde se van haciendo comparaciones sucesivas de las imágenes actuales con el fotograma inmediato anterior, de esta manera se logra disminuir en gran medida los efectos de distorsión, mala calidad, cambios de iluminación y otros de la imagen, puesto que al tomar dos imágenes adyacentes captadas entre el mínimo instante de tiempo detectable por la cámara se minimizan al máximo las diferencias entre las características locales.

4.3 Seguimiento del target de fotograma a fotograma

La propuesta del seguimiento de un target basado en la detección de fotograma a fotograma se presenta en el diagrama de la Figura 52.

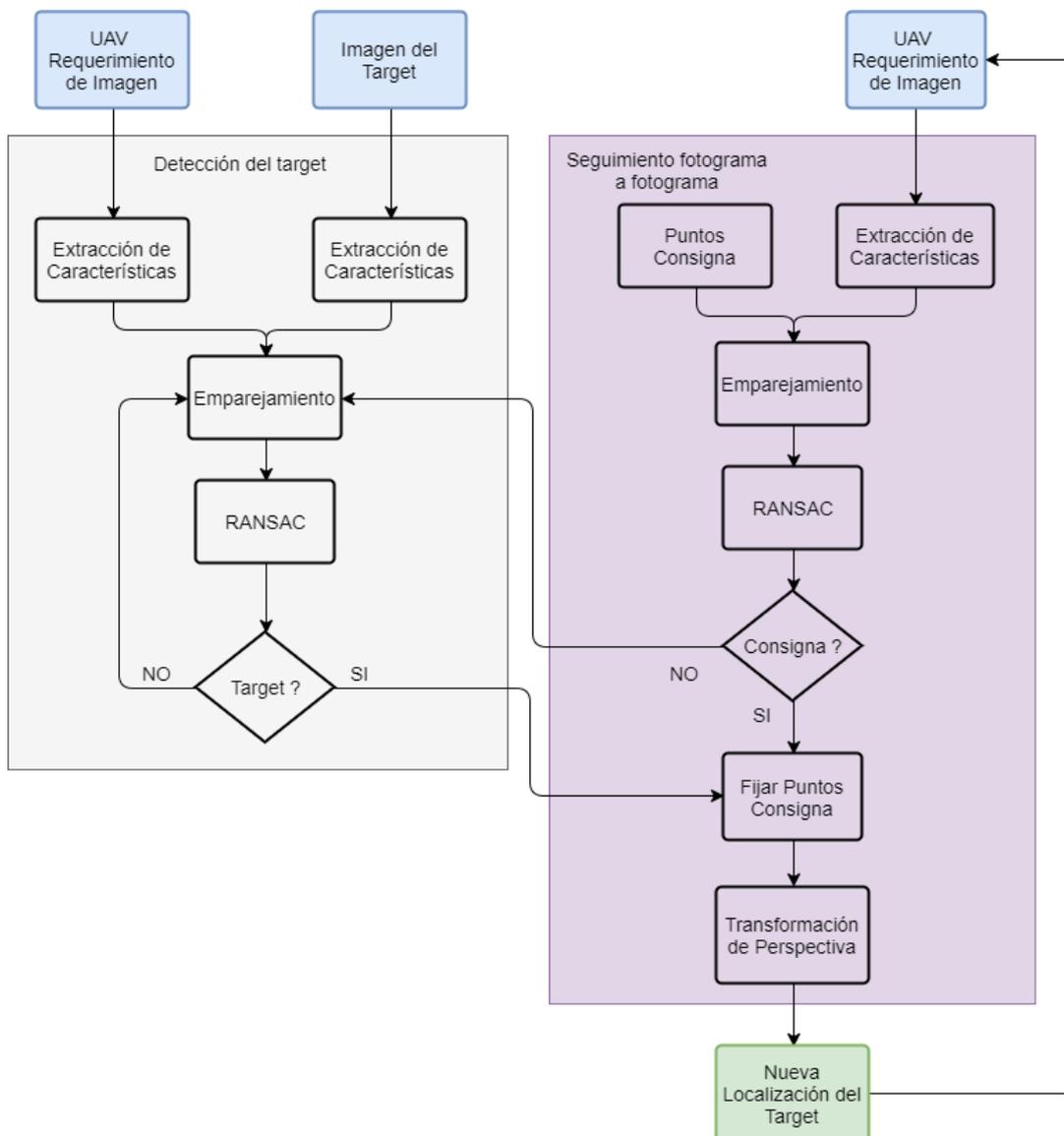


Figura 52. Diagrama de Flujo de la propuesta de detección y seguimiento visual de fotograma a fotograma.

4.3.1 Proceso de detección y seguimiento

Para realizar el seguimiento de un target entre fotogramas consecutivos como se observa en el diagrama de la Figura 52 se toma el concepto de (Aguilar & Angulo Bahón, 2013).

Para la ejecución del procedimiento descrito se deben considerar dos etapas:

- Detección inicial del target
- Asignación de la consigna

En la primera etapa se realizan los procedimientos descritos anteriormente Figura 44 donde se toman los puntos de interés correspondientes al target, posteriormente se comparan con los puntos de interés de cada fotograma hasta superar el umbral que permita etiquetar la detección del target en el fotograma correspondiente con un recuadro que lo identifique visualmente Figura 50.

Una vez que se ha comprobado la detección del target y se ha asignado un recuadro que da la ubicación al mismo en el fotograma correspondiente, se realiza la asignación de las nuevas consignas que son:

- Las características locales del fotograma actual como nuevas consignas para la evaluación de correspondencias en el siguiente fotograma.
- El recuadro que evidencia la localización del target como consigna para la siguiente interacción de la transformación geométrica.

La homografía aplicada no tiene transformaciones muy significativas por el tiempo de captura de la cámara haciendo más precisa la traslación del recuadro.

Como resultado existen más puntos coincidentes entre las imágenes como se puede observar en la Figura 53 dándole una característica que no se podía lograr antes y es que no se pierde el target ya que la escena tiene suficientes puntos de interés para lograr detecciones satisfactorias. Como se puede apreciar en la Figura 53 los puntos superan considerablemente el umbral del target por lo cual se usará un umbral referencial de 100 puntos para la correspondencia entre fotogramas consecutivos.

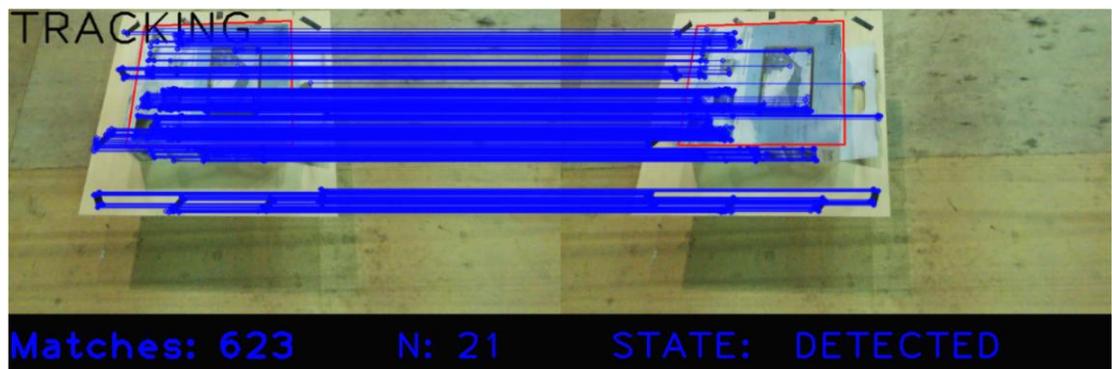


Figura 53. Correspondencia de puntos de interés entre fotogramas consecutivos.

Este proceso funciona correctamente haciendo de la detección y seguimiento del target con una cámara en movimiento más robusto al tener más puntos que logran ser emparejados, no obstante, el error acumulado puede influenciar desfavorablemente a la hora de obtener su ubicación dado que el recuadro consigna puede sufrir pequeños desplazamientos producto del ruido en el proceso de caracterización de imágenes que al acumularse provocan transformaciones geométricas inexactas y por ende cambios de la posición y deformaciones que no coinciden con las correspondientes al target como se puede observar en la secuencia de imágenes de la Figura 54.

En la Figura 54 se exponen seis muestras de la detección de 948 fotogramas consecutivos en los cuales la participación de los puntos de interés correspondientes al fondo de la imagen y los movimientos alrededor de la misma provocan cambios geométricos acumulados que se pueden ver como los desplazamientos y deformaciones del recuadro.

Por simplicidad las imágenes y pruebas a continuación se realizan únicamente graficando los resultados del emparejamiento de características en una transformación geométrica, es decir solo con el recuadro de la imagen del target sin incluir la unión de correspondencias entre fotogramas.

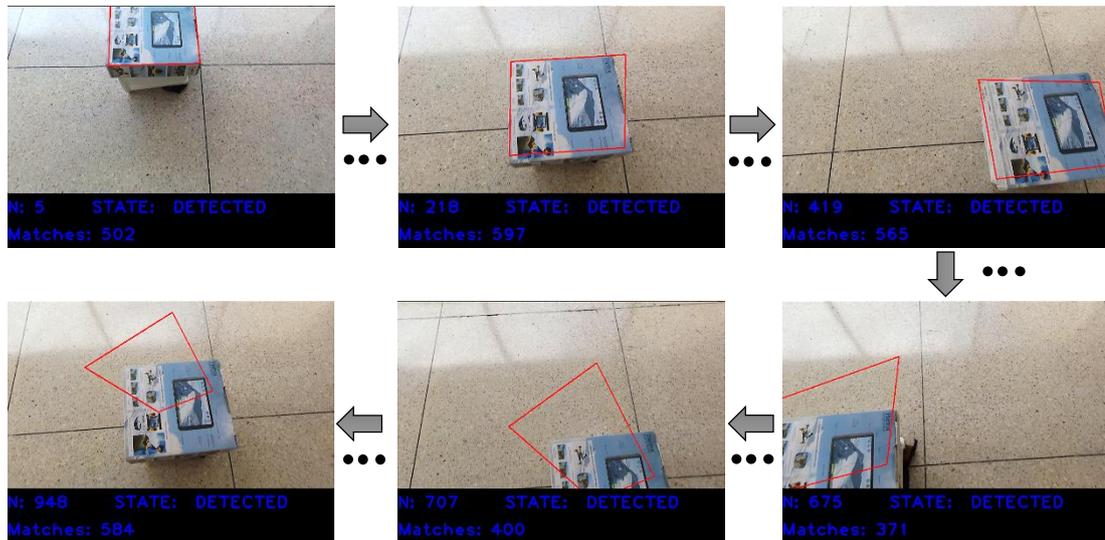


Figura 54. Secuencia de imágenes con deformaciones del recuadro.

4.3.2 Refrescamiento de la posición del target

Debido al error acumulado de la Figura 54 se recurre al refrescamiento periódico del target como etapa final para el seguimiento robusto de un objetivo.

El refrescamiento del target se debe realizar cada intervalo de tiempo “ t ” de manera que el recuadro de la localización del target no alcance a sufrir cambios significativos.

Para fines prácticos y mejor rendimiento se ha determinado realizar el refrescamiento cada 0.2 segundos (10 fotogramas). El refrescamiento consiste en hacer una extracción de características locales al espacio ocupado por el recuadro más un umbral de error, de manera que se analiza un espacio pequeño logrando el emparejamiento de puntos de interés entre el target y la porción de imagen correspondiente.

Luego de haber encontrado las correspondencias del target dentro de la porción de imagen correspondiente se realiza la estimación de la homografía para aplicar la

transformación de perspectiva a los puntos del recuadro del target consigna obteniendo el recuadro en la posición exacta del fotograma analizado.

Para ejemplificar en la Figura 55 se ha tomado un ciclo de refrescamiento del target cada 30 fotogramas pudiendo así apreciar la corrección del recuadro después del ciclo de refrescamiento.

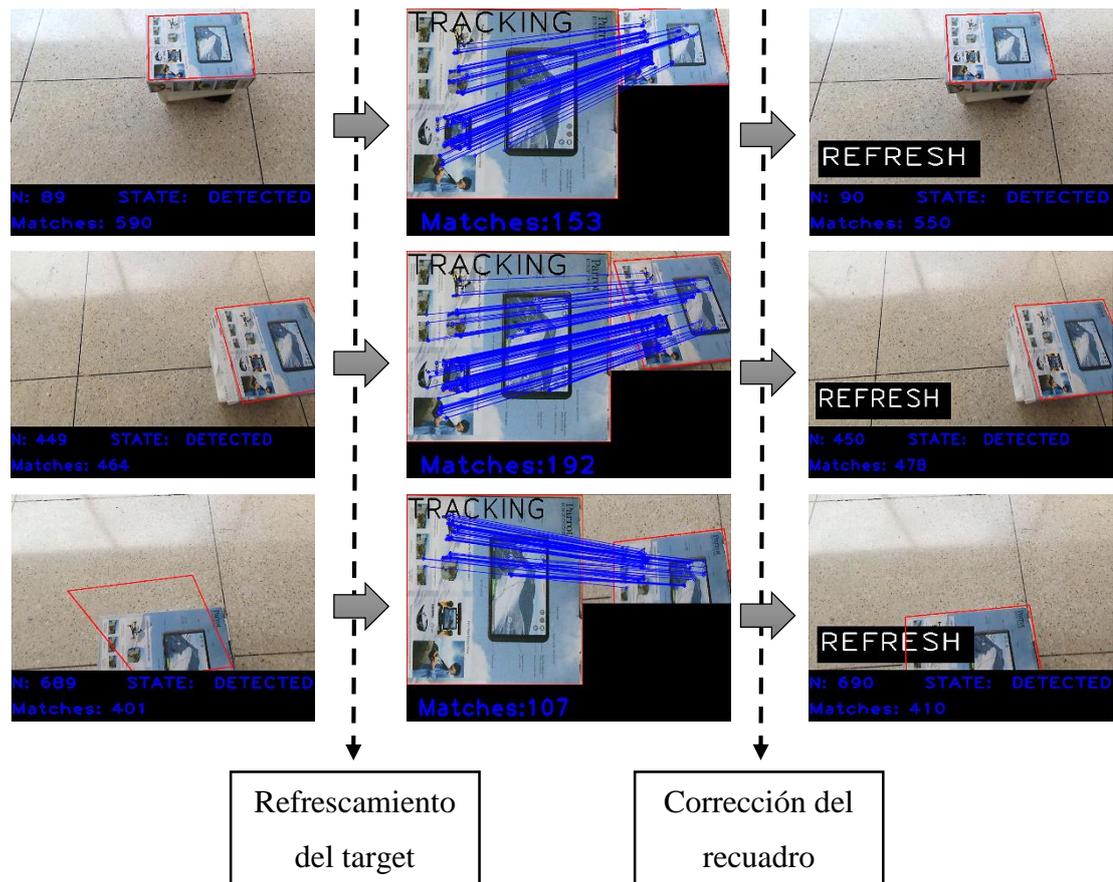


Figura 55. Secuencia de imágenes con refrescamiento del target y corrección del recuadro. (Izquierda) fotograma anterior al refrescamiento, (centro) porción de imagen que contiene al target en la escena, (derecha) fotograma siguiente con corrección del recuadro.

El diagrama de flujo del proceso de seguimiento entre fotogramas consecutivos con refrescamiento se presenta en la Figura 56.

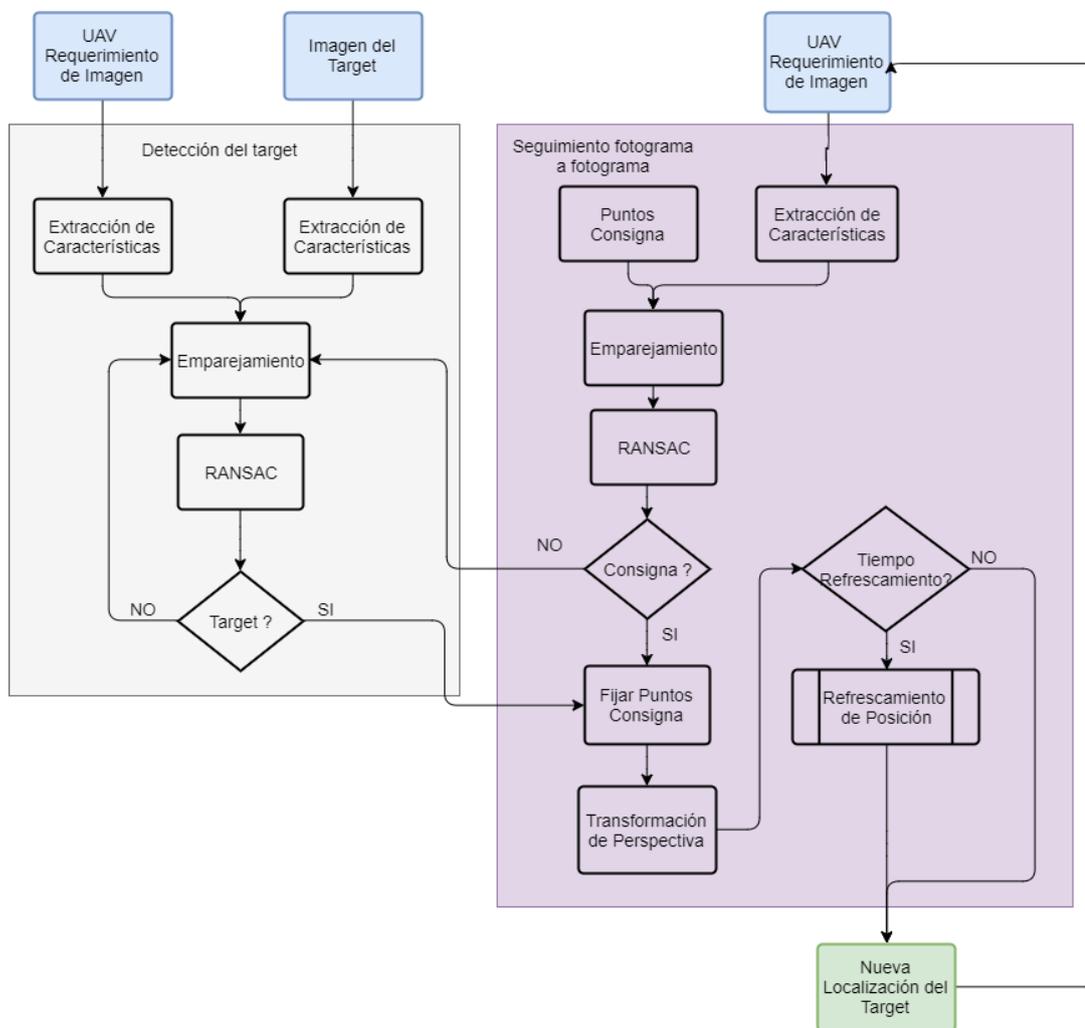


Figura 56. Diagrama de flujo del seguimiento del target entre fotogramas consecutivos con refrescamiento periódico de la posición.

4.3.3 Posición virtual del target

El siguiente problema a abordar con el procedimiento antes descrito es la pérdida de vista del target ocasionada por cambios repentinos de velocidad en el seguimiento o perturbaciones externas que alejen de la vista directa al target lo que ocasiona que no se pueda seguir su trayectoria o acercarse a la misma. Para solucionar este problema se ha realizado un refrescamiento de la “**posición virtual del recuadro**” del target con el algoritmo antes descrito de correspondencia entre fotogramas consecutivos.

Se aplica la transformación de perspectiva al recuadro para desplazarlo hacia donde se ha movido la imagen con respecto al fotograma anterior lo que permite obtener una posición virtual del target sin la necesidad de verlo directamente, la Figura 57 muestra un esquema de la secuencia de interés que representa la pérdida de vista del target.

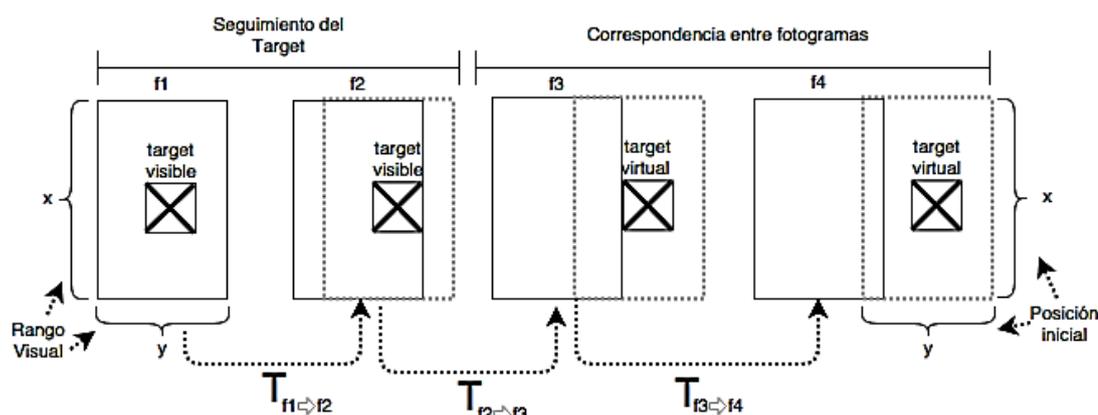


Figura 57. Target virtual representado por transformaciones entre fotogramas consecutivos.

Se puede apreciar que la operación $T_{f_1 \rightarrow f_2}$ representa la transformación geométrica de perspectiva del fotograma f_1 al f_2 donde se aplica la transformada de la ecuación (35) en los puntos del target que estuvo originalmente en f_1 para pasarlos a una nueva posición en f_2 .

De esta manera se puede recuperar la posición dejando a la vista nuevamente el target si se compensa el movimiento realizado como una estimación del error de posición del recuadro en comparación con la posición de la cámara frente al fotograma visible como se aprecia en la Figura 58.

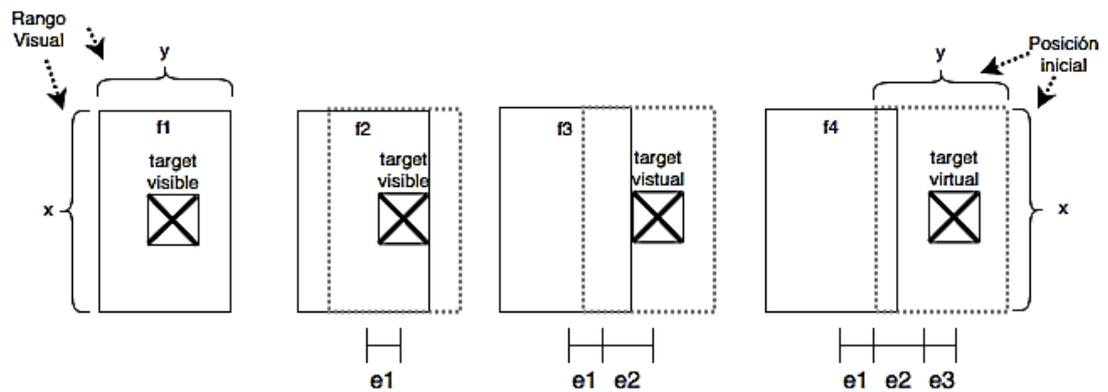


Figura 58. Movimientos respecto al target representados como un error de posición.

El error de un frame f_i con respecto al frame f_{i-1} se representa en la ecuación (48):

$$e(i) = \Delta(x, y)_{f_{i-1} \rightarrow f_i} \quad (48)$$

Donde la posición visual del target está definida por la sumatoria de los errores relativos a cada frame respecto al anterior como se muestra en la ecuación (49):

$$TGT_{x,y}(i) = \sum_{k=0}^{N-1} e_{x,y}(i - k) \quad (49)$$

Con $N = \text{número de frames}$ y $e(0) = 0$ representando al error nulo en la posición inicial.

CAPITULO V

SEGUIMIENTO Y ATERRIZAJE DEL VEHÍCULO AEREO NO TRIPULADO

En este capítulo se presentan las propuestas que se han planteado para lograr el aterrizaje del dron en una plataforma en movimiento, empezando por el enfoque de modelo de movimientos del dron basado en estimación de movimiento entre fotogramas consecutivos, posteriormente se presenta el diseño e implementación de un controlador clásico para la compensación del movimiento requerido a fin de disminuir la distancia entre el dron y la plataforma para finalmente realizar el descenso a velocidad constante y aterrizar el dron si se encuentra en un umbral definido.

5.1 Estimación del modelo de movimiento del dron

La plataforma utilizada para este trabajo es el dron “Bebop 1” como se explica en el Capítulo III por su bajo costo, dimensiones pequeñas, cualidades de video y seguridad de vuelo. Para poder controlar al dron recurriendo a la teoría de control clásico hace falta en primer lugar estimar el modelo de movimiento del mismo, para ello se han considerado los beneficios y limitaciones que ofrece.

5.1.1 Consideraciones para el modelado

La estimación del modelo dinámico de movimiento de un dron es analizada en gran medida en la literatura de manera indirecta, con un modelo matemático basado en las formulaciones de Euler Lagrange y otros enfoques (Bohdanov, 2012; Guenard, Hamel, & Mahony, 2007; Ortiz González, 2014; Parra Muñoz et al., 2013). Adicionalmente existen varios enfoques para la estimación directa del modelo de movimiento de un UAV de dimensiones pequeñas utilizando las acciones de control como entradas del sistema y las medidas de la IMU (Unidad de medición inercial) como salida del mismo (Aguilar & Angulo, 2016; Aguilar, Casaliglla, & Polit, 2017; Aguilar, Casaliglla, & Pólit, 2017; Aguilar, Casaliglla, Pólit, Abad, & Ruiz, 2017), en los enfoques anteriores se coincide en que el movimiento a lo largo de los ejes (x, y, z) están vinculados significativamente a cada una de las acciones de control Tabla 13:

Tabla 13
Acciones de control vinculadas al movimiento en cada eje del sistema de coordenadas.

Acción de control	Rotación Z	Eje X	Eje Y	Eje Z
Pitch	Involuntario	Voluntario	Involuntario	Involuntario
Roll	Involuntario	Involuntario	Voluntario	Involuntario
Yaw	Voluntario	Involuntario	Involuntario	Involuntario
Altitud	Involuntario	Involuntario	Involuntario	Voluntario

Fuente: (Aguilar & Angulo, 2016)

Considerando el movimiento voluntario provocado por las acciones de control en cada uno de los ejes, se puede aproximar la planta a un conjunto de modelos SISO (Una entrada y una salida) donde la acción de control se ve reflejada en movimientos a lo largo del eje correspondiente.

Con la ayuda de estas premisas, se ha recurrido a las acciones de control que se especifican en la Tabla 5 de los movimientos en Pitch, Roll, Yaw y Altitud para la generación de movimientos en cada uno de los ejes asociando únicamente los movimientos voluntarios con las ordenes de control.

Posteriormente al analizar las cualidades operativas del dron con respecto a su modelamiento basado en mediciones inerciales se han encontrado los inconvenientes que se detallan a continuación:

- El modelamiento basado en mediciones inerciales depende completamente de la fidelidad de los datos obtenidos durante el proceso.
- Para la presente aplicación (Aterrizaje automático en superficies móviles basado en seguimiento de puntos de interés) se requiere que la retroalimentación del estado del dron sea tanto o más rápida que el procesamiento de imágenes, dado que las acciones de control se centran en la velocidad del dron.

- El drone Bebop de Parrot transmite datos de navegación al controlador de ROS a una tasa aproximada de 5 Hz lo que hace de la captación de datos para el modelamiento y control un proceso inexacto por su baja frecuencia.

Por los inconvenientes presentados se ha recurrido a la estimación del movimiento acumulado basado en visión artificial y transformación afín para poder realizar el modelamiento del drone.

5.1.2 Modelamiento basado en estimación del movimiento entre imágenes

La estimación de movimiento acumulado basado en transformación afín es una propuesta presentada en el trabajo de (Aguilar & Angulo, 2014, 2016) y utilizado posteriormente para el modelamiento de un drone en (Aguilar, Salcedo, Sandoval, & Cobeña, 2017) obteniendo resultados muy parecidos a los que se consiguen al realizar el modelamiento directo basado en mediciones inerciales.

La propuesta para el modelamiento consiste en extraer las características locales de las imágenes obtenidas durante el vuelo para poder compararlas y determinar el comportamiento del sistema Figura 59. El modelamiento se realiza “off board”, es decir, se registra la secuencia de video durante el vuelo para su procesamiento posterior.

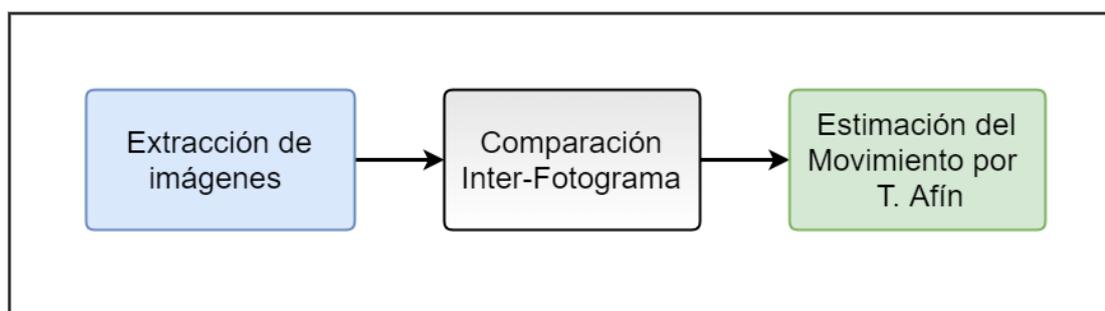


Figura 59. Bloques de la obtención de parámetros de movimiento del drone.

El proceso de comparación se realiza desde la segunda imagen capturada y se comparan las posiciones de las características emparejadas entre las dos imágenes I_0 e I_1 para extraer una transformación que se ajuste al movimiento de la plataforma, movimiento que se presenta como una transformación geométrica en las imágenes Figura 60.

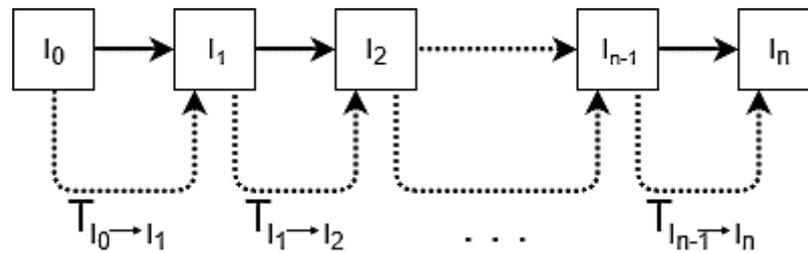


Figura 60. Transformaciones geométricas entre imágenes consecutivas.

De manera que entre un par de imágenes I_0 e I_1 se puede establecer que la deformación entre ellas se define como una transformación de sus puntos.

Para la elección de la transformación geométrica a aplicarse en la correspondencia de puntos se debe considerar:

- La forma de la ventana en el espacio tridimensional. Dado que los cambios de perspectiva en el espacio provocan distorsiones que no pueden ser abordadas por transformaciones con pocos grados de libertad. Dependiendo del plano de visión se pueden usar diferentes transformaciones desde la similitud no reflectiva hasta la homografía en función de la complejidad.
- El tipo de movimiento generado. Los cambios de perspectiva constantes en la imagen provocan transformaciones más complejas y sensibles al ruido.

Para solventar estos problemas (Ma et al., 2003) discute la utilización de la transformación affine cuando se trata de deformaciones correspondientes a un plano de imagen paralelo al plano de la cámara, definiendo que en superficies relativamente pequeñas y planas los cambios del punto de vista en dicha imagen se pueden traducir en distorsiones affine con una escala asociada.

Para este trabajo el plano de imagen no es paralelo al plano de la cámara pues la toma se realiza a -70° en Pitch, teniendo 20° de inclinación del plano de imagen con respecto al piso o plano de tierra Figura 61, no obstante en (Ma et al., 2003) se generaliza el caso en que todas las correspondencias en una ventana de la imagen se mueven aproximadamente hacia una misma dirección donde se puede considerar nuevamente la aproximación del movimiento entre imágenes como una distorsión affine con una escala asociada a pesar de no ser una toma por completo cenital.

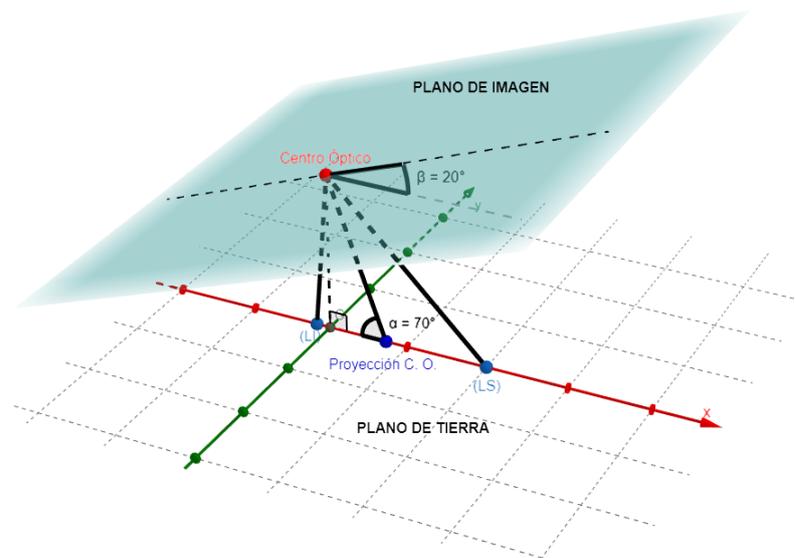


Figura 61. Levantamiento del plano de imagen con respecto al plano de tierra $\beta = 20^\circ$.

Para la extracción de los parámetros de movimiento del drone basado en transformación afín se toman en cuenta las siguientes consideraciones Tabla 14:

Tabla 14

Consideraciones para la extracción de los parámetros de movimiento del drone.

Premisa	Valor
Captura de imágenes	Inclinación a -70°
Rango de Movimientos	X, Y (Pitch y Roll)
Altura	Constante (un metro)
Dirección del movimiento	Única en lapsos de tiempo
Características del terreno	Plano

5.1.3 Obtención del modelo a partir de los datos acumulados entre imágenes

De igual manera que en (Aguilar, Salcedo, Sandoval, et al., 2017) para estimar el modelo matemático de movimiento del drone se ha tomado un video a bordo de una secuencia dada de movimientos que responden a las órdenes de control enviadas desde la estación en tierra mediante el controlador “bebop_autonomy”, para delimitar la respuesta con respecto a los parámetros de movimiento se han enviado acciones de control únicamente para pitch y roll excluyendo la variación de altura y la rotación.

El lugar donde se ha realizado el modelamiento es el coliseo de la Universidad de las Fuerzas Armadas ESPE.

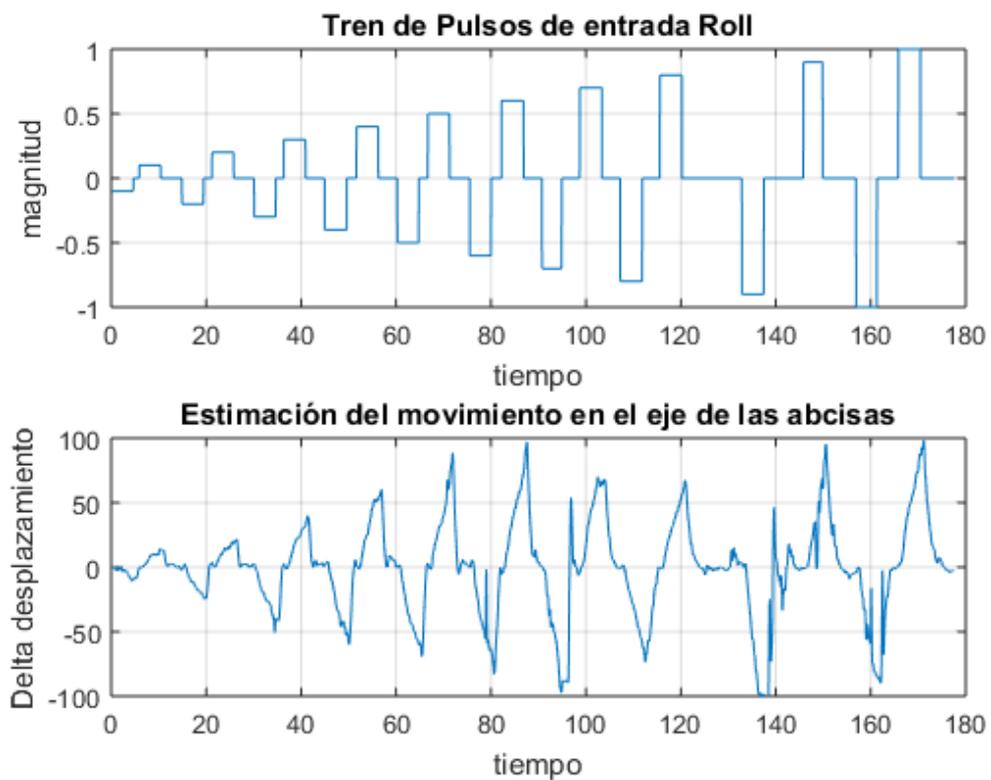
Para realizar la toma del video se ha tomado las imágenes con una inclinación de -70° , posteriormente se envía un tren de pulsos con las diferentes velocidades permisibles para el drone en pitch y roll en la escala de $[-1 \dots 1]$ (Figura 62 y Figura 63 superior) espaciados con un paso de 0.1 entre cada pulso y manteniendo una altura constante de un metro.

Posteriormente se extraen las características de cada par de fotogramas consecutivos como ha sido explicado en el capítulo anterior y se estima la transformación affine entre las correspondencias con la función de OpenCV “getAffineTransform” que retorna una matriz de 2 filas por 3 columnas, una versión simplificada de la matriz de la ecuación (31).

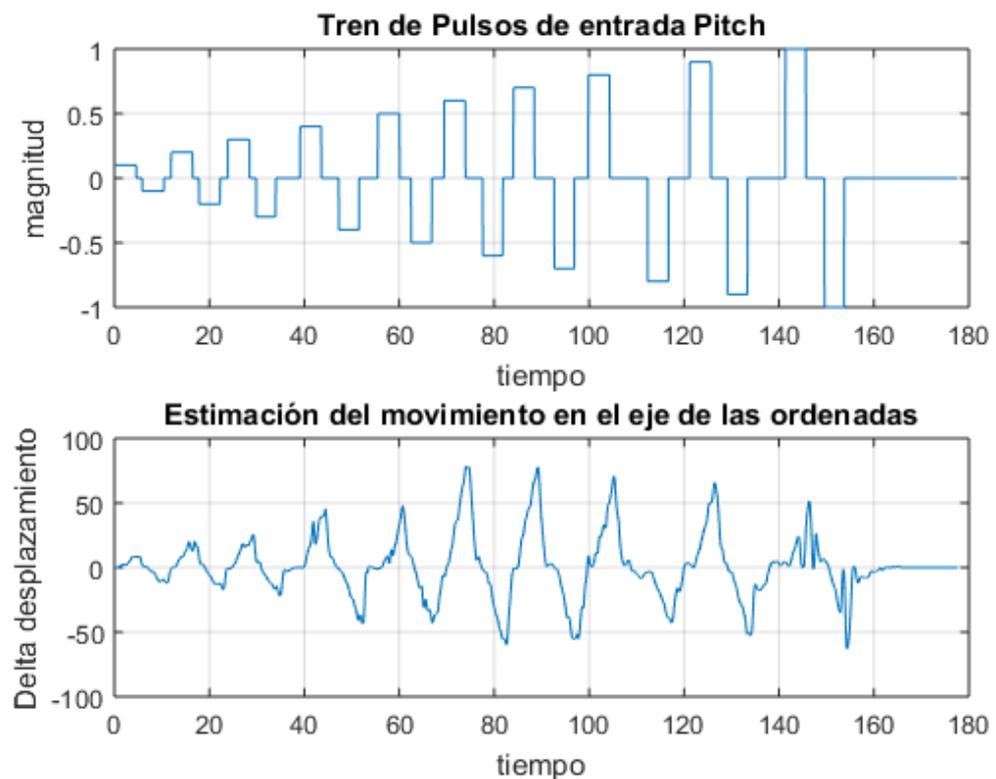
Los parámetros de traslación en x, y son fáciles de extraer pues están en la tercera columna de la matriz, como se aprecia en la ecuación (50):

$$A = \begin{pmatrix} a & b & t_x \\ c & d & t_y \end{pmatrix} \quad (50)$$

En la Figura 62 y Figura 63 (inferior) se puede apreciar los valores filtrados del delta de desplazamiento estimado por la transformación affine en cada instante de tiempo.



**Figura 62. Comportamiento del dron con órdenes de control en Roll.
Superior: entrada. Inferior: salida.**



**Figura 63. Comportamiento del dron con órdenes de control en Pitch.
Superior: entrada. Inferior: salida.**

Observando las gráficas del comportamiento del drone para las ordenes de control se puede apreciar que a partir de los pulsos respectivos de $[0.5, -0.5]$ en Roll y $[0.6, -0.6]$ en Pitch hacia adelante, el delta de desplazamiento presenta valores estacionarios no representativos. Este fenómeno se puede dar por las altas velocidades alcanzadas por el drone que hacen que se distorsione la imagen perdiendo la confiabilidad en la detección. Por este motivo se truncará en dichos valores las características del sistema en Pitch y Roll.

Para la estimación de la planta se ha utilizado la herramienta de Matlab “procest” que permite describir la dinámica de un sistema o planta a partir de datos temporales, los datos de entrada son las señales de control y los de salida son los “delta de desplazamiento” en cada caso.

La selección del tiempo de muestreo se realiza en función de la frecuencia de la cámara del drone la cual según el parámetro “VideoFramerate” del conjunto “PictureSettings” Tabla 4 es de 25 fotogramas por segundo, obteniendo la función de transferencia:

$$G(s) = \frac{K_p}{1 + T_p s} \quad (51)$$

Con una ganancia de $K_p = 110.59$, y constante de tiempo $T_p = 1.6128$ para Roll y de $K_p = 96.339$, y constante de tiempo $T_p = 1.84$ para Pitch.

En la Figura 64 y Figura 65 se muestran las comparaciones del modelo estimado con los datos reales para Roll y Pitch respectivamente.

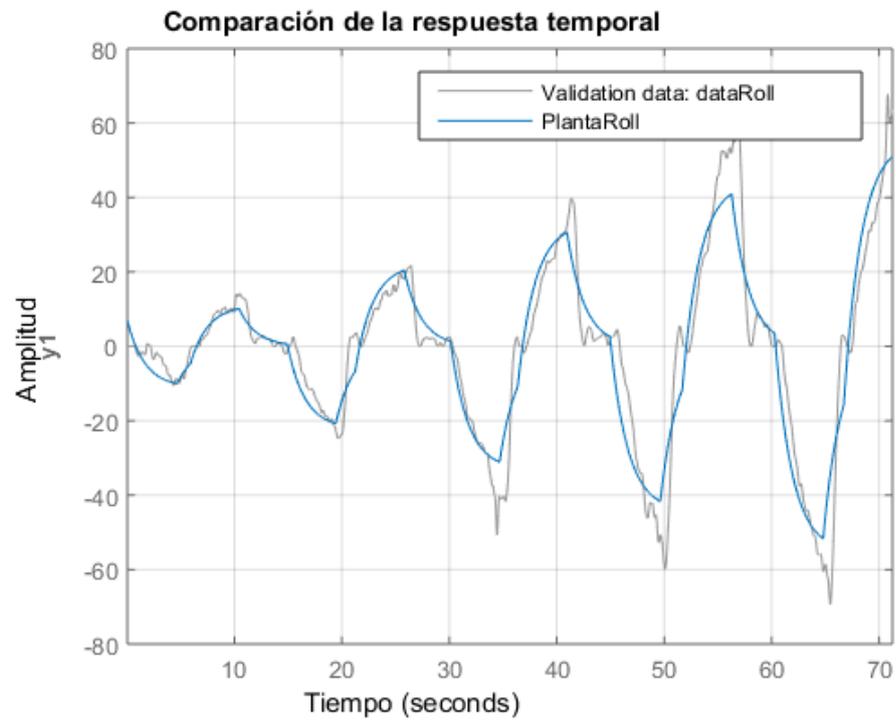


Figura 64. Identificación de la Planta en Roll.

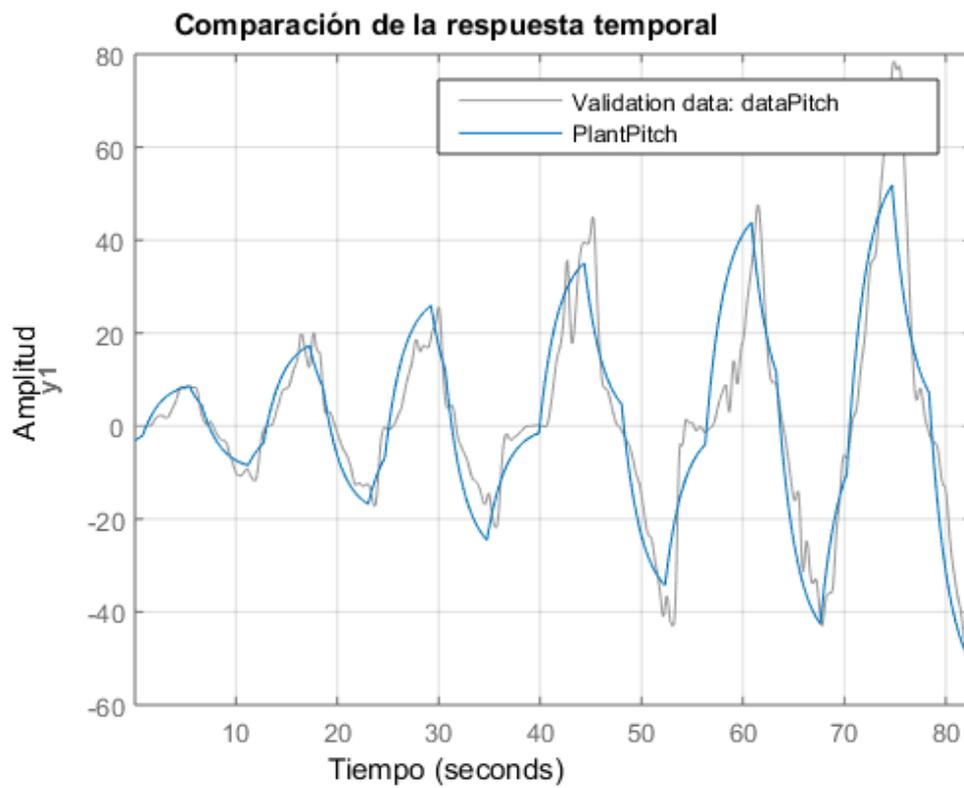


Figura 65. Identificación de la Planta en Pitch.

5.2 Diseño del controlador

Para poder realizar el aterrizaje automático del drone en la plataforma móvil se implementa un controlador que permite asegurar el correcto seguimiento del target. El esquema de control se puede apreciar en la Figura 66.

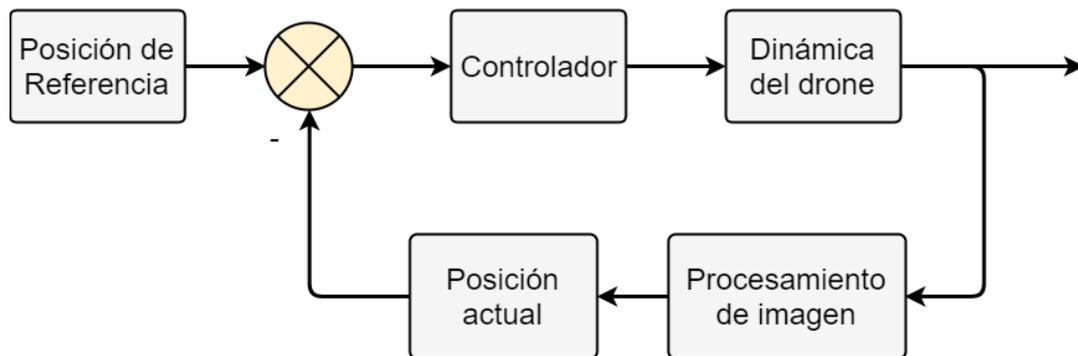


Figura 66. Esquema de control para el seguimiento del target mediante la realimentación de la posición visual.

Para el diseño del controlador se ha utilizado la planta obtenida en el paso anterior y el método de ubicación de los polos para determinar las ganancias de un controlador PID de la forma:

$$G_c(s) = \left(\frac{s^2 K_d + s K_c + K_i}{s} \right) \quad (52)$$

Donde los valores K_c , K_i y K_d son las ganancias, el controlador $G_c(s)$ está representado en su forma paralela.

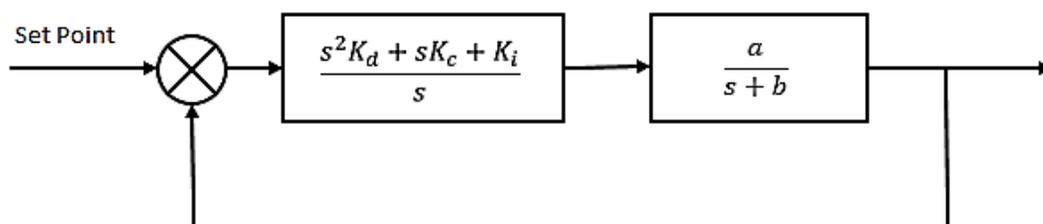


Figura 67. Esquema de diseño del controlador.

Para Roll se ha determinado que el tiempo de estabilización es $t_s < 4 \text{ seg}$, $M_p = 0\%$ dada la respuesta de la planta, estableciendo un polo en $s = -0.9$ para el diseño del controlador en tiempo continuo.

Para el controlador en Pitch se debe considerar que las acciones de control deben ser menos agresivas que en Roll ya que el rango de visión de 856×480 no permite tener un rango amplio de control en el eje de las ordenadas, por ello, para el diseño del controlador en tiempo continuo se ha establecido un tiempo menor a 5 seg y, $M_p = 0\%$ posicionando un polo en $s = -0.7$

Los polinomios característicos correspondientes son:

$$\Delta_1(s) = s^2 + bs + a(s^2K_d + sK_c + K_i) \quad (53)$$

$$\Delta_2(s) = (s + p_s)^2 = p_0s^2 + p_1s + p_2 \quad (54)$$

Realizando la relación:

$$\Delta_1(s) = \Delta_2(s) \quad (55)$$

La solución matricial del polinomio se resuelve como $PC = D$ donde:

$$\begin{aligned} s^2: \quad 1 + K_d a &= p_0 \\ s^1: \quad b + aK_c &= p_1 \\ s^0: \quad aK_i &= p_2 \end{aligned} \quad (56)$$

Las matrices P, C y D son:

$$\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} \begin{bmatrix} K_d \\ K_c \\ K_i \end{bmatrix} = \begin{bmatrix} p_0 - 1 \\ p_1 - b \\ p_2 \end{bmatrix} \quad (57)$$

El reemplazo de valores para Roll con un polo en $p_s = -0.9$ y los valores $a = 68.57$ y $b = 0.62$ de la Planta en Roll de la ecuación (51) deja las matrices de la ecuación (57) de la siguiente manera:

$$\begin{bmatrix} 68.57 & 0 & 0 \\ 0 & 68.57 & 0 \\ 0 & 0 & 68.57 \end{bmatrix} \begin{bmatrix} K_d \\ K_c \\ K_i \end{bmatrix} = \begin{bmatrix} 0 \\ 1.18 \\ 0.81 \end{bmatrix} \quad (58)$$

Obteniendo las ganancias respectivas de:

$$K_c = 0.017; \quad K_i = 0.0118; \quad K_d = 0$$

El controlador le permite al sistema eliminar al máximo el error de estado estable llegando a alcanzar el desplazamiento deseado en el eje de las abscisas en un tiempo de 2.74 segundos como se observa en la respuesta al impulso unitario del sistema en Roll Figura 68.

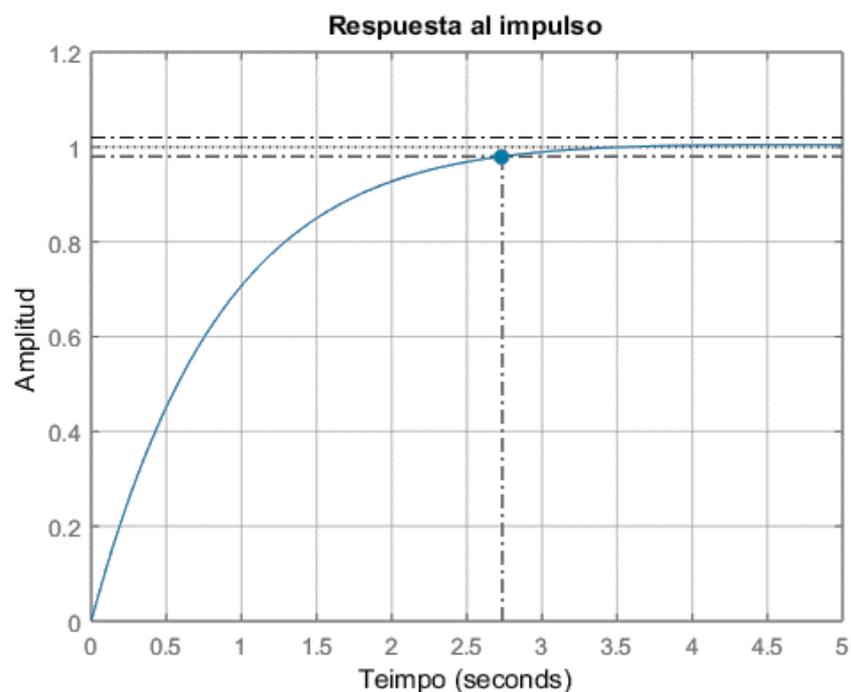


Figura 68. Respuesta al impulso unitario del sistema controlador planta en Roll.

Posteriormente el reemplazo de valores para Pitch con un polo en $p_s = -0.7$ y los valores $a = 52.34$ y $b = 0.54$ de la Planta en Pitch de la ecuación (51) deja las matrices de la ecuación (57) de la siguiente manera:

$$\begin{bmatrix} 52.34 & 0 & 0 \\ 0 & 52.34 & 0 \\ 0 & 0 & 52.34 \end{bmatrix} \begin{bmatrix} K_d \\ K_c \\ K_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0.8567 \\ 0.49 \end{bmatrix} \quad (59)$$

Obteniendo las ganancias respectivas de:

$$K_c = 0.016; \quad K_i = 0.009; \quad K_d = 0$$

Un controlador que reduce el error en estado estable del desplazamiento deseado en el eje de las ordenadas para el control en Pitch con un tiempo de estabilización de 4.11 segundos como se puede ver en la Figura 69.

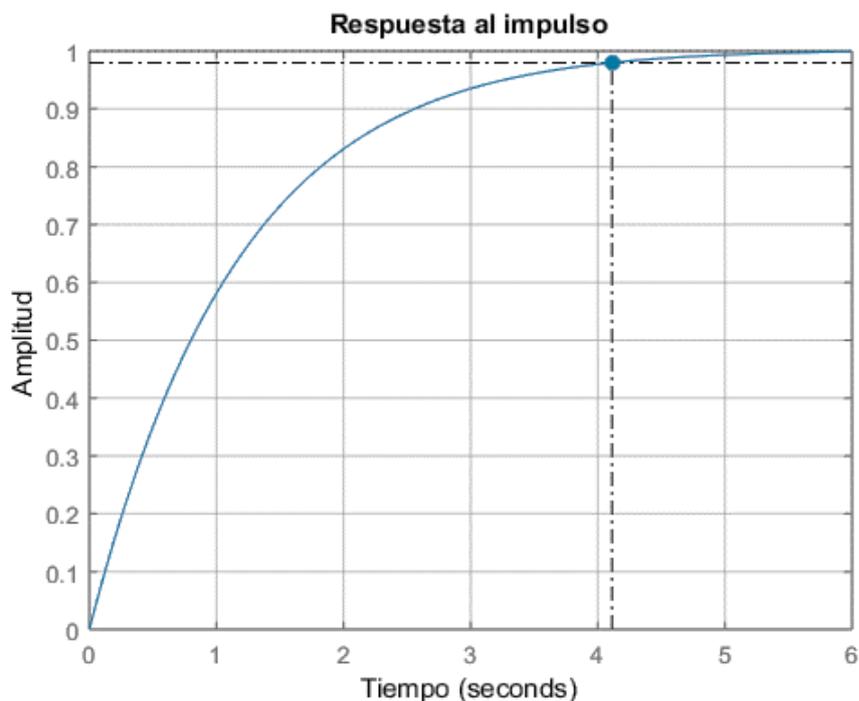


Figura 69. Respuesta al impulso unitario del sistema controlador planta en Pitch.

5.3 Seguimiento y aterrizaje del UAV en la plataforma móvil terrestre

En el Capítulo IV se realizó la detección del target basada en puntos de interés, luego de discriminarlo del resto de la escena se procedió a representar su posición con un recuadro, la detección se realizó con gran efectividad al aplicar la propuesta planteada de seguimiento visual entre fotogramas consecutivos con refrescamiento de la posición del target. Para lograr el aterrizaje dentro de la plataforma se implementa un controlador que permite el seguimiento y compensación del movimiento requerido para poder aterrizar dentro de un umbral al momento del descenso. A continuación se presenta el diagrama de flujo del sistema de aterrizaje Figura 70 seguido por el detalle de los procedimientos que involucra.

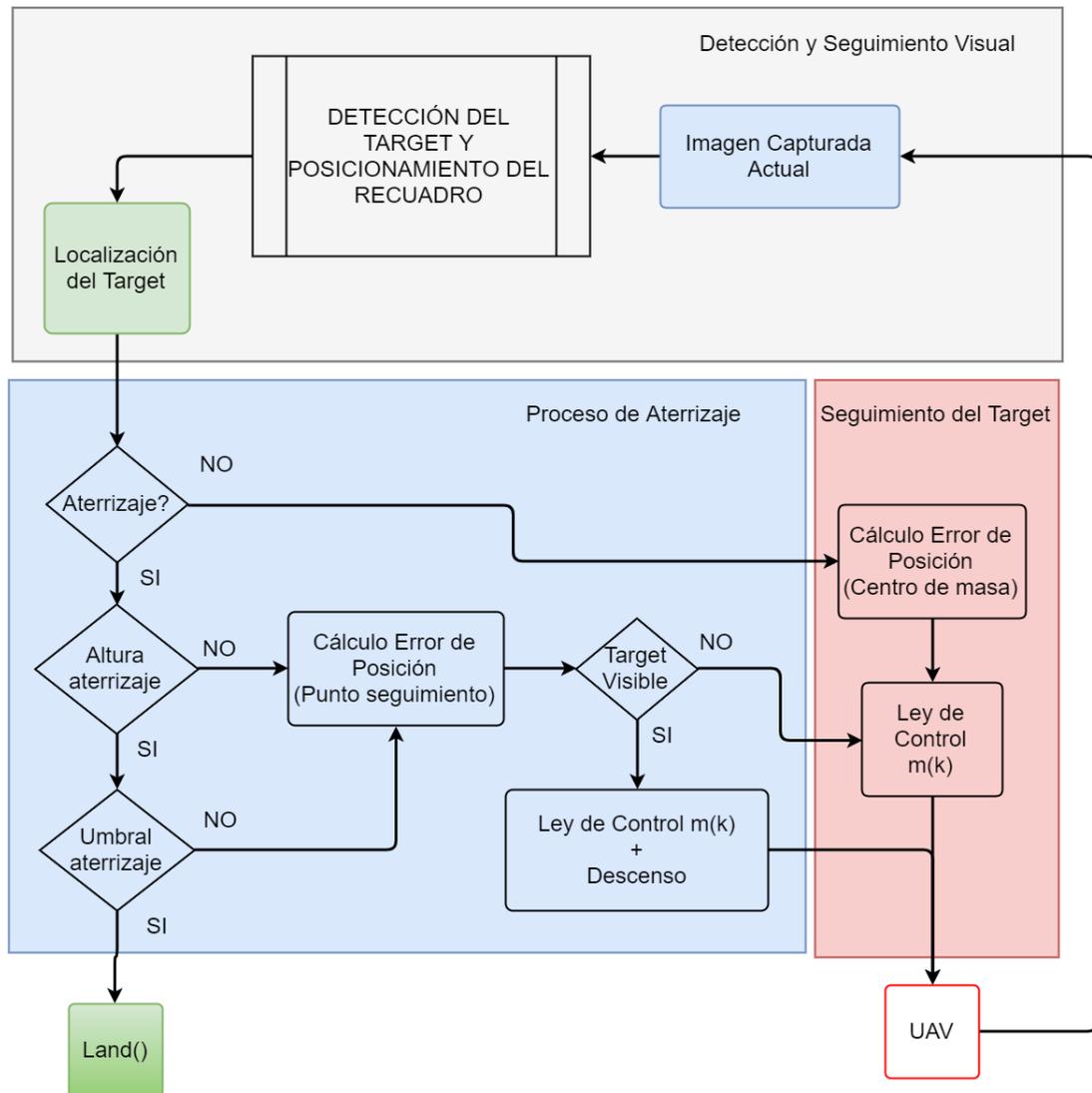


Figura 70. Diagrama de flujo del aterrizaje automático en superficies móviles.

5.3.1 Implementación del controlador para seguimiento de la plataforma móvil terrestre

A continuación para realizar el seguimiento de la plataforma móvil, se ha implementado un controlador digital para el modelo PID paralelo de la ecuación (52), donde la salida “m” en el intervalo discreto “k” del controlador es:

$$m(k) = m(k - 1) + K_c \times (e(k) - e(k - 1)) + K_i \times T_s \times e(k) + \frac{K_d}{T_s} \times (e(k) - 2e(k - 1) + e(k - 2)) \quad (60)$$

Algoritmo 1.1. Algoritmo del Controlador implementado

if $t = 0$

Calcular acción de control con valores pasados de acción y control iguales a cero

Registrar valores del control actual y error actual

end

if $t > 0$

Calcular acción de control

Registrar valores del control actual y errores actual y pasado

end

En el algoritmo descrito la acción de control es representada por la ecuación (60) y los valores de control y error pasados son cero. Se incluyen dos archivos de controlador al programa principal; uno del controlador con los parámetros para Roll y otro para Pitch.

El “set point” para cada uno de los controladores es una función proporcional al error de posición del centro de masa del target con respecto al centro de la imagen, es decir:

$$SP_{Roll} = K_r \times E_x \quad (61)$$

$$SP_{Pitch} = K_p \times E_y \quad (62)$$

Se utiliza el centro de masa del target debido a las distorsiones ocasionadas por la inclinación de la cámara con respecto al plano de tierra donde transita el vehículo.

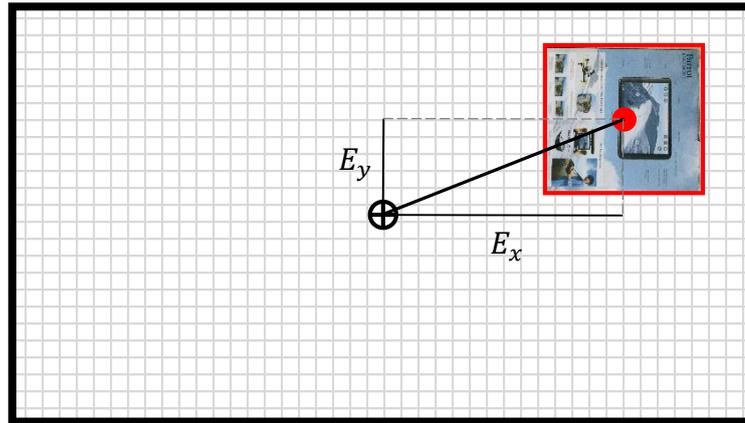


Figura 71. Error de posición del centro de masa del target con respecto al centro de la imagen.

El centro de masa del target se calcula usando los momentos centrales del recuadro que lo contiene, al tener solo cuatro elementos es un proceso rápido y está definido por:

$$mu_{ij} = \sum_{x,y} [(x, y) \cdot (x - \bar{x})^j \cdot (y - \bar{y})^i] \quad (63)$$

$$\bar{x}_{tg} = \frac{m_{10}}{m_{00}}, \quad \bar{y}_{tg} = \frac{m_{01}}{m_{00}} \quad (64)$$

De esta manera se obtienen el centro del recuadro, OpenCV lo realiza mediante la función “moments()”, finalmente los valores del error de posición en X, Y; E_x y E_y respectivamente se obtienen de la resta de los centros del target menos el centro de la imagen, ambos en pixeles.

$$E_x = \bar{x}_{tg} - 856/2 \quad (65)$$

$$E_y = \bar{y}_{tg} - 428/2 \quad (66)$$

Dado que los errores E_x y E_y se encuentran en pixeles, es necesario tomar en cuenta la escala a la cual se asocian para relacionarlos con distancias fijas utilizables para el control de posición del dron.

5.3.1.1 Influencia de la altura como escala asociada

Una de las consideraciones más importantes a la hora de realizar el modelado del sistema fue la elección de la transformación geométrica que traduciría las imágenes cambiantes a lo largo del tiempo en movimientos del drone, para ello se determinó usar la transformación affine dándole al modelo un comportamiento asociado a una escala.

En este caso la escala asociada termina siendo la altura dado que la toma de imágenes se realiza aproximadamente de manera cenital.

Tomando en consideración que la escala asociada a la toma de imágenes en movimiento influye en la distancia recorrida que se puede reconocer mediante una transformación geométrica entonces se debe realizar un ajuste a dichas medidas para diferentes alturas, en el caso del descenso o ascenso del drone.

Para ello se tomará un simple factor de escala dependiente de la distancia de detección, siendo este una función del ángulo de apertura de la cámara y la altura de la toma. El cálculo del factor de escala se realiza siguiendo las siguientes consideraciones:

- El centro óptico de la cámara está a la misma altura que el drone sobre el piso.
- Los ángulos de apertura de la cámara son: 50° en el eje de las ordenadas y 80° en el eje de las abscisas.
- Como el error de posición se obtiene de una medición con respecto al centro de la imagen, se tomarán las distancias máximas de detección en la proyección del centro óptico sobre el plano de tierra como una aproximación a todas las distancias.

En la Figura 72 y Figura 73 se pueden apreciar las proyecciones de la apertura de la cámara en el eje de las ordenadas y abscisas (Pitch y Roll) respectivamente para la detección a la altura de un metro sobre el plano de tierra.

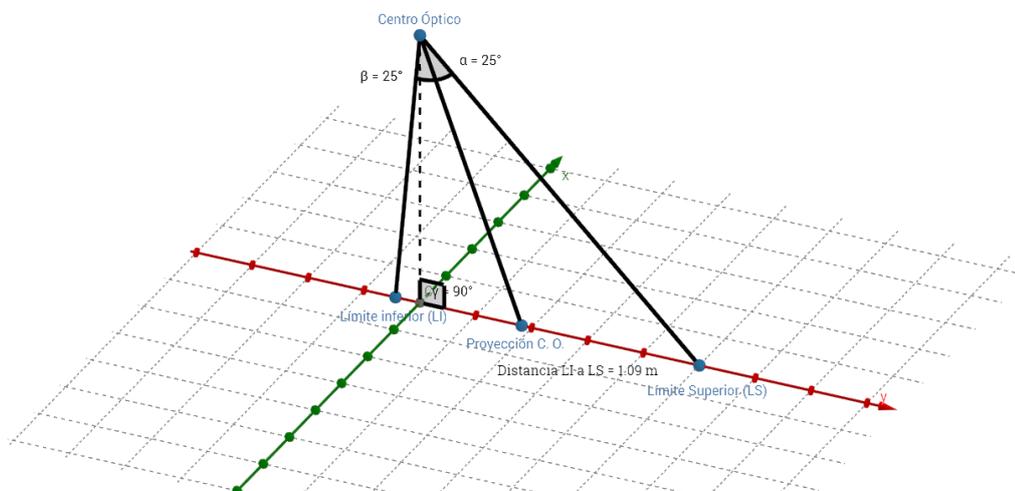


Figura 72. Longitud de detección del eje de las ordenadas sobre el plano de tierra a la altura de un metro.

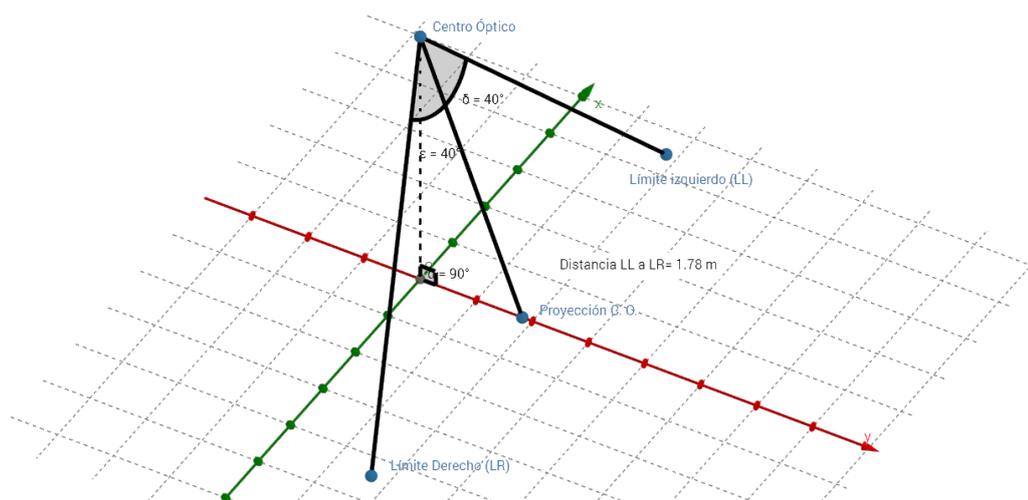


Figura 73. Longitud de detección del eje de las abscisas sobre el plano de tierra a la altura de un metro.

El cálculo se realiza relacionando el ancho y alto de la imagen con la distancia estimada de detección. En ambos ejes se relacionan los ángulos con la altura para determinar las distancias mediante razones trigonométricas en las cuales intervienen los ángulos de apertura de la cámara como se indica en las ecuaciones (67) y (68).

$$\text{Distancia en Y} = \text{Altura} \times (\tan(45) + \tan(5)) \quad (67)$$

$$\text{Distancia en X} = \frac{\text{Altura}}{\cos(90-70)} \times (2 \times \tan(40)) \quad (68)$$

En la ecuación (67) se puede evidenciar la apertura de 50° en el eje de las ordenadas (Pitch) y en la ecuación (68) la apertura de 80° en el eje de las abscisas (Roll).

Para un metro de altura la *DistanciaY* es 1.09 metros y la *DistanciaX* 1.78 metros como se puede observar en la Figura 72 y Figura 73 respectivamente.

Finalmente como el modelo de movimiento y los controladores para Pitch y Roll están basados en las mediciones de la cámara a un metro de altura entonces esta altura se usa como referencia en la relación:

$$\#Pixeles_{Pitch} = \frac{\#Pixeles_{ventana_Pitch} \times DistanciaY(Altura_{actual})}{DistanciaY(Altura_{referencia})} \quad (69)$$

$$\#Pixeles_{Roll} = \frac{\#Pixeles_{ventana_Roll} \times DistanciaX(Altura_{actual})}{DistanciaX(Altura_{referencia})} \quad (70)$$

5.3.2 Seguimiento de la plataforma terrestre

Para disminuir errores introducidos en la detección del target y su posterior seguimiento se ha definido un umbral con respecto al centro de la imagen a partir del cual se efectúa el control del movimiento del dron.

El umbral se ha definido tomando en cuenta la tolerancia para el posicionamiento del dron dentro de la plataforma de aterrizaje (Capítulo III Figura 34) manteniendo siempre a la proyección del centro óptico de la cámara dentro de la misma. La tolerancia de la Figura 34 en el eje horizontal de la imagen (abscisas) es de 15 cm y 18 cm en el eje vertical (ordenadas), aplicando la ecuación (69) y (70) se obtiene la cantidad de pixeles del umbral en el eje vertical y horizontal de la imagen respectivamente.

$$\#Pixeles_{Roll} = \frac{856 \text{ px} \times 0.15 \text{ m}}{1.78 \text{ m}} = 72.13 \text{ px} \quad (71)$$

$$\#Pixeles_{Pitch} = \frac{480 \text{ px} \times 0.18 \text{ m}}{1.09 \text{ m}} = 79.26 \text{ px} \quad (72)$$

Estos valores representan el 8.42% de la ventana en horizontal y el 16.51% en vertical, debido a que el ángulo de visión es mayor en el eje horizontal se aumentará el umbral en ese eje al 10% dando 86 px un equivalente a 17 cm en la plataforma.

A continuación se muestra un ejemplo del seguimiento de la plataforma terrestre en una trayectoria rectilínea y a velocidad constante de la misma, las condiciones de la prueba son:

- Trayectoria rectilínea
- Tiempo de seguimiento 6 segundos (150 interacciones)
- Espacio interior (dentro de los hangares del CICTE)
- Velocidad de la plataforma: 0.5 metros por segundo
- Altura de detección con un umbral de 1.1 a 1.3 metros

En la Figura 74 se puede apreciar una secuencia de 8 imágenes separadas mostrando el seguimiento del target desde su detección inicial hasta el tiempo determinado de seguimiento.

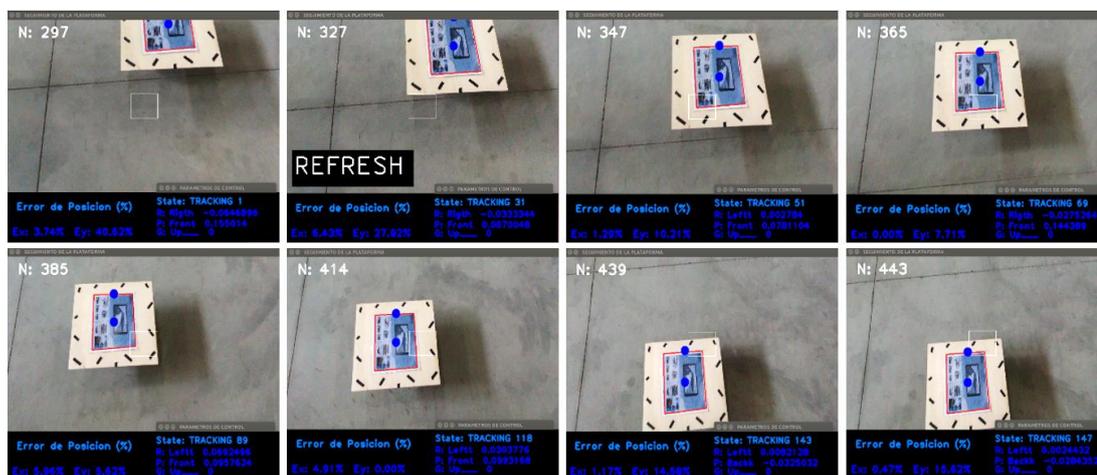


Figura 74. Secuencia de imágenes de la detección y seguimiento de la plataforma terrestre.

En la posición inferior izquierda de las imágenes de la figura anterior se puede observar el error de posición porcentual con respecto al centro de la ventana y a la derecha las acciones de control de Roll, Pitch y Altitud (R, P, G) derivadas del controlador ante

los errores. En la Figura 75 y Figura 76 se puede apreciar las traslaciones del dron y su trayectoria en el tiempo de detección.

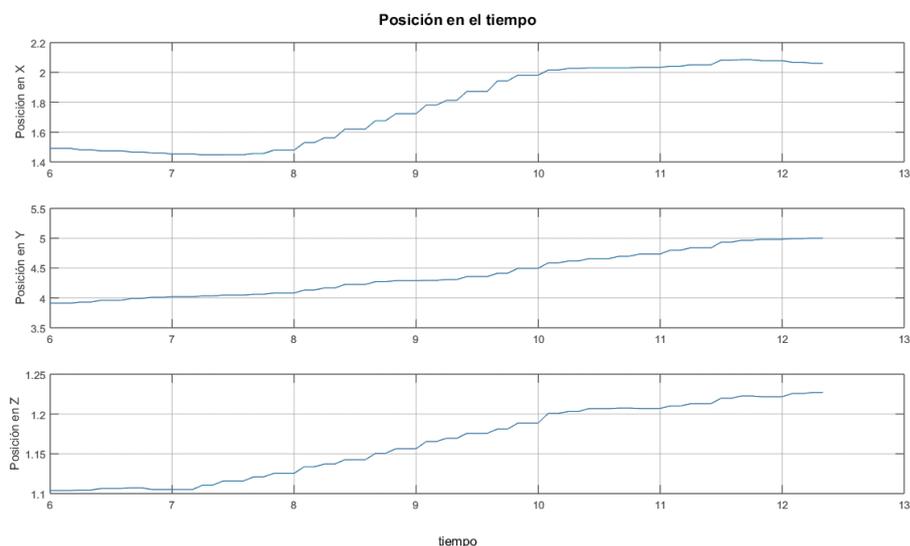


Figura 75. Posición del dron en cada instante de tiempo desde la detección inicial hasta la final.

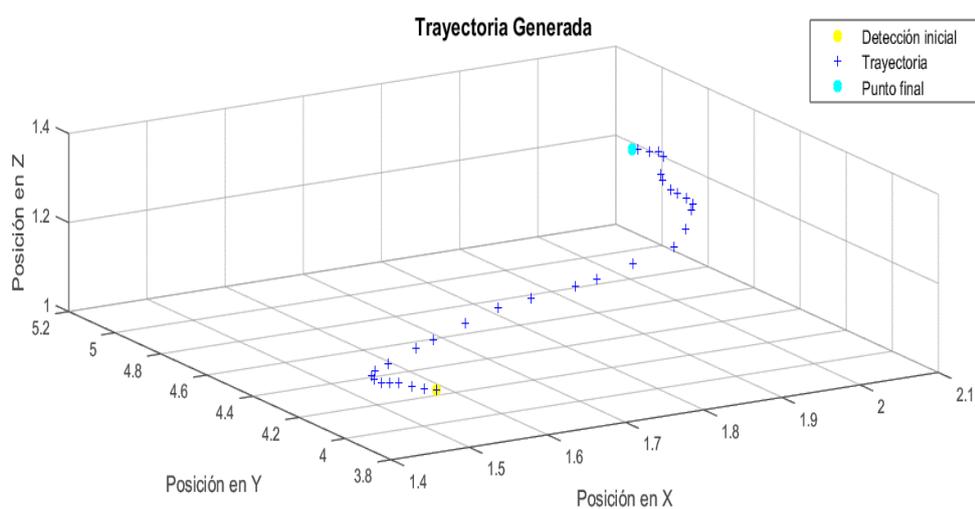


Figura 76. Trayectoria recorrida por el dron en el proceso de seguimiento.

Los datos de las figuras anteriores han sido extraídos de la Odometría transmitida por el dron, cabe mencionar que los datos del dron son propensos a inexactitudes, especialmente la Odometría al ser una fusión de datos inerciales con los de la cámara manteniendo errores apreciables en distancias pequeñas, no obstante a pesar de dichas inexactitudes un filtrado de los datos permite mostrar el comportamiento del dron durante el tiempo de seguimiento.

5.3.3 Aterrizaje del UAV en la plataforma móvil terrestre

Una vez realizado el seguimiento de la plataforma mediante el control de posición en los ejes (x, y) se debe realizar el control de altura para el descenso del UAV, para ello se han obtenido los datos de altura del drone mediante el subscriptor explicado anteriormente (Capítulo III) al parámetro de altitud.

5.3.3.1 Descenso y aterrizaje sobre la plataforma

El descenso es la etapa final del aterrizaje, para determinar un aterrizaje correcto se necesita que todas las etapas anteriores funcionen correctamente desde la detección del target hasta el seguimiento de la trayectoria. La etapa final del aterrizaje se ha realizado con las siguientes premisas:

- La detección de la plataforma de aterrizaje se realiza de fotograma a fotograma con un refrescamiento de la posición del target cada 10 interacciones
- Se utiliza el controlador PID del seguimiento del target para disminuir la distancia estimada de aterrizaje.
- La velocidad de descenso es constante.
- El error es estimado basándose en la escala asociada.
- La orden final de aterrizaje se realiza mediante el comando Land() (por los beneficios presentados en el Capítulo III, sección 3.4.2) del programa de control del driver (implementado en el Capítulo III) que disminuye la velocidad de todos los motores del drone hasta que toca la base, esto se realiza cuando se ha acercado lo suficiente a la plataforma de aterrizaje.

Además de las premisas mencionadas anteriormente se debe tomar en cuenta que para asegurar el posicionamiento correcto del drone sobre el target hay que evaluar si se encuentra en una zona correcta para el descenso. Dado que el controlador de movimientos del drone permite su desplazamiento por ambos ejes se ha planteado un conteo de “N” interacciones donde el target se mantiene en su totalidad dentro del rango visual del drone, superado ese rango de interacciones se empieza la secuencia de aterrizaje.

La secuencia de aterrizaje lleva consigo que la velocidad en el eje Z se fija constante y es en descenso, además solamente se desciende en caso de tener a la vista el target.

Se fija un nuevo punto de seguimiento que en este caso no es el centro del recuadro del target sino su perpendicular superior dentro del recuadro como se aprecia en la Figura 77 haciendo que los giros de la plataforma sean irrelevantes para el aterrizaje pues siempre busca el punto superior al recuadro indistintamente de la orientación. Esto se hace debido a las dimensiones del drone que al proyectar su centro óptico en la parte superior del target se posiciona aproximadamente en el centro de la plataforma.

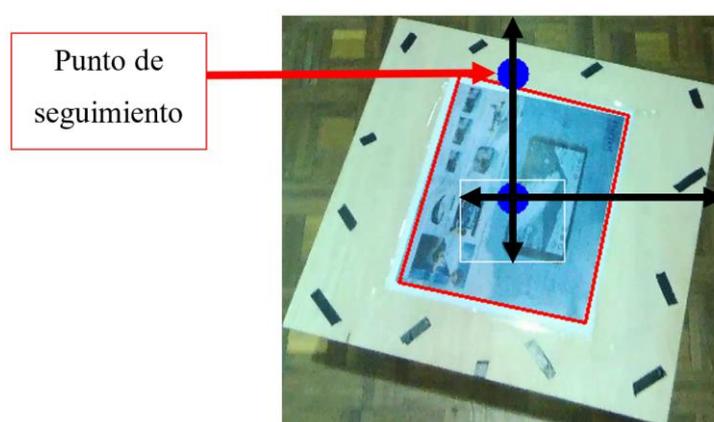


Figura 77. Fijación del nuevo punto como centro para seguimiento en la parte superior del recuadro.

Finalmente al descender se comprueba la detección del target dentro de la imagen y se evalúa la posición que mantiene el drone con respecto a la plataforma, la condición de aterrizaje es:

- La proyección del centro óptico de la cámara está dentro de un umbral establecido dentro de la plataforma de aterrizaje.
- La distancia vertical de la plataforma al drone es menor a 40 cm.

Cuando se cumplen esas dos condiciones se envía finalmente la instrucción de Land() que permite el aterrizaje seguro del drone y mantiene la posición como se mencionó en el Capítulo III.

El umbral de aterrizaje mencionado se ha establecido de la siguiente manera:

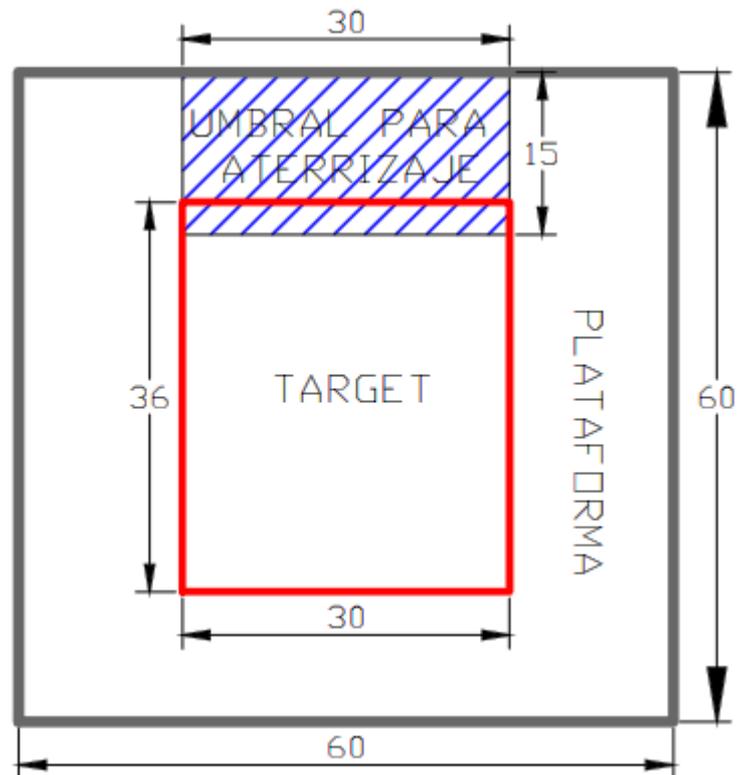


Figura 78. Umbral de aterrizaje definido (las medidas se encuentran en cm).

EL umbral para aterrizaje es donde se debe proyectar el centro óptico de la cámara del drone, es decir el centro de la imagen debe estar contenido en el recuadro de umbral.

El umbral ha sido definido tomando en cuenta la tolerancia del posicionamiento del drone dentro de la plataforma (Figura 34) ya que a una altura de 40 cm la máxima distancia en Y está definida por la distancia del centro óptico hacia su proyección en el eje Y (ordenadas):

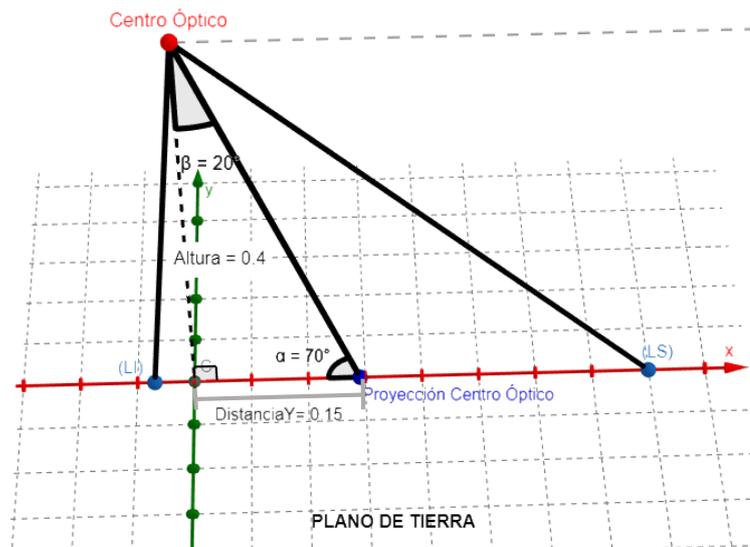


Figura 79. Proyección del centro óptico de la cámara del dron a una altura de 40 cm y 70° de descenso de la cámara virtual.

$$\begin{aligned}
 \text{Distancia } Y &= \text{Altura} \times \tan(20^\circ) \\
 &= 0.4 \times \tan(20^\circ) \\
 &= 0.1456 \text{ m}
 \end{aligned}
 \tag{73}$$

De manera que para estar dentro del umbral de aterrizaje la cámara del dron está ubicada como máximo 14.56 cm detrás de la plataforma y el centro del dron 30.56 cm detrás del borde inferior del umbral en la plataforma.

La inclinación de la cámara virtual solo se presenta en el eje de las ordenadas de modo que la proyección en el eje de las abscisas corresponde aproximadamente al mismo punto.

Tomando en cuenta que el dron mide 32 cm de largo y 28 de ancho su centro está en 16 cm y 14 cm. El posicionamiento del dron dentro de la plataforma se puede definir de la siguiente manera:

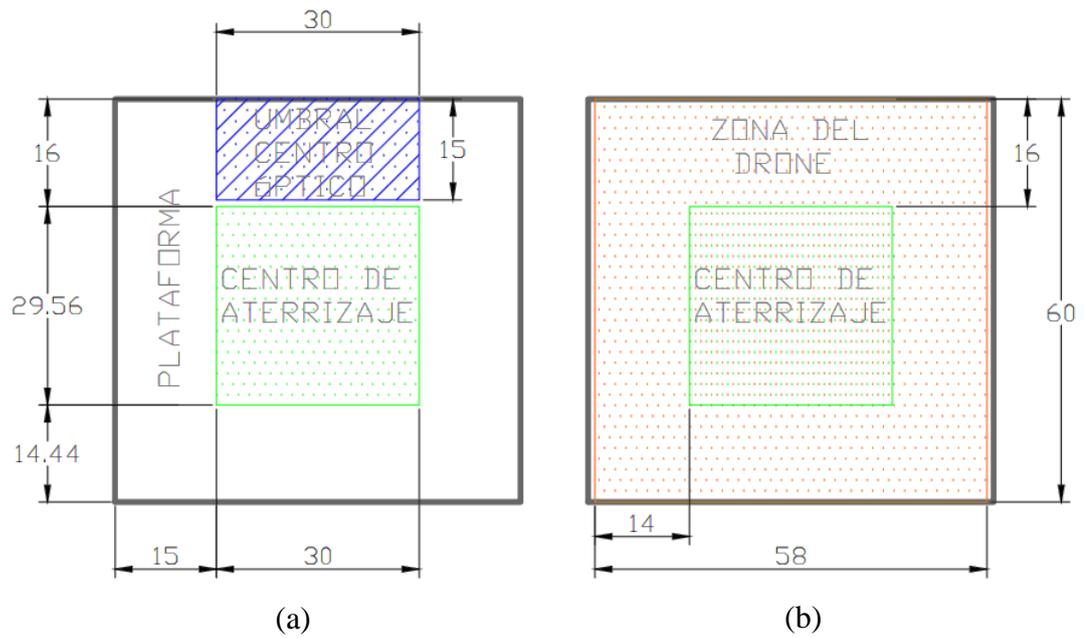


Figura 80. Zona de posicionamiento del drone debido al umbral definido. (a) Zona del centro del drone. (b) Zona máxima ocupada por todas las dimensiones del drone.

La secuencia de imágenes de la Figura 81 muestran el seguimiento y aterrizaje del drone en la plataforma móvil para $N = 150$ interacciones después de haber empezado a moverse la plataforma terrestre y velocidad de descenso de 0.3 m/s .

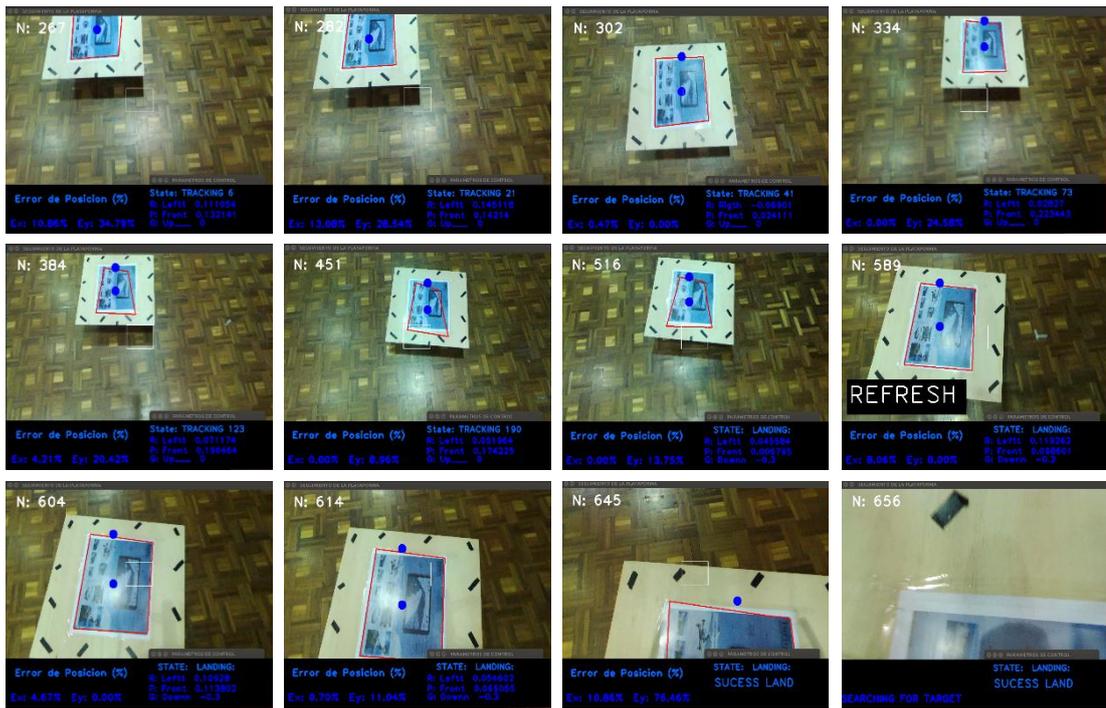


Figura 81. Secuencia de imágenes desde la detección del target hasta el aterrizaje dentro de la plataforma.

En las figuras a continuación se puede apreciar la posición del drone en cada instante de tiempo desde el despegue hasta el aterrizaje final, así como su trayectoria.

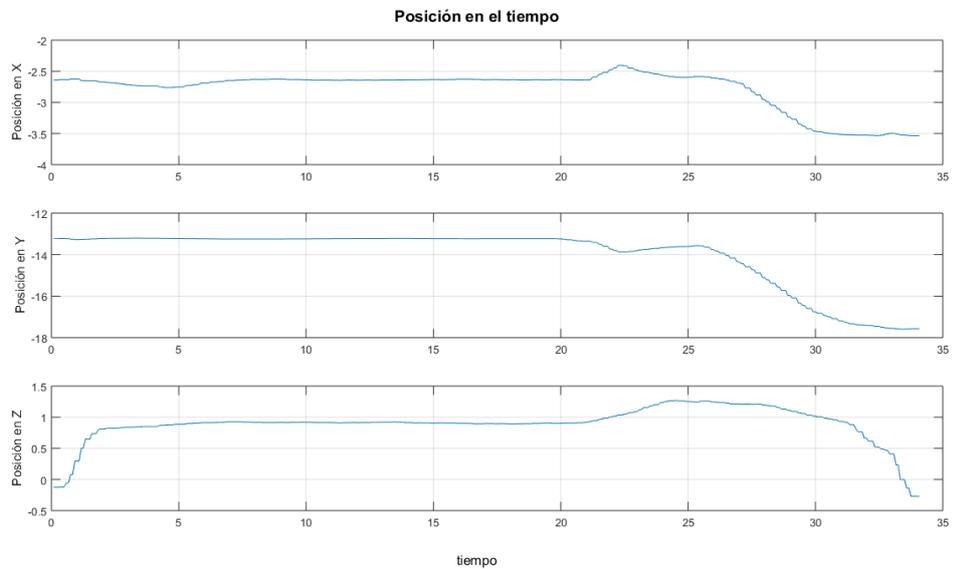


Figura 82. Posición del drone en cada instante de tiempo desde el despegue hasta el aterrizaje.

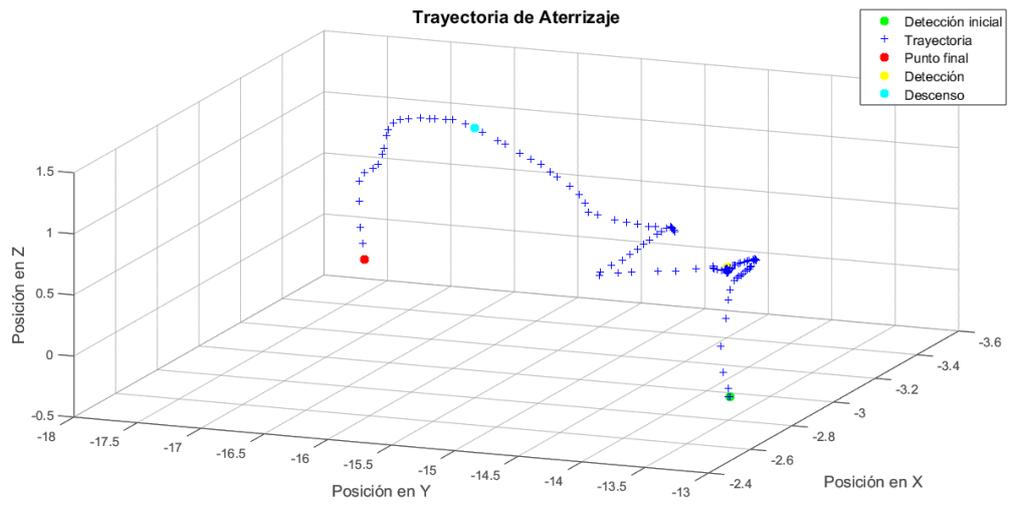


Figura 83. Trayectoria recorrida por el dron en el proceso de aterrizaje.

CAPITULO VI

PRUEBAS EXPERIMENTALES DE FUNCIONAMIENTO Y RESULTADOS

Para evaluar el desempeño de la propuesta de aterrizaje automático de un UAV de este trabajo, se ha planteado un conjunto de pruebas experimentales para diferentes escenarios y condiciones de interés del entorno, los cuales se detallan más adelante.

Debido a la naturaleza del aterrizaje basado en visión por computador se han planteado dos campos a evaluarse en cada escenario que dependen cada uno del buen desempeño del otro y son:

- Detección y seguimiento visual.
- Control del movimiento.

Una de las condicionales más importantes al realizar mediciones basadas en visión por computador es la variabilidad del entorno, para abarcar un campo amplio se han realizado un conjunto de pruebas en tres entornos diferentes:

- **Hangares del Centro de Investigación Científica y Tecnológica del Ejército (CICTE).** Está ubicado dentro de las instalaciones de la Universidad de las Fuerzas Armadas – ESPE y es un espacio interior parcialmente cerrado donde se ha autorizado previamente realizar las pruebas por el DIRECTOR DEL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA DEL EJÉRCITO (CICTE).
- **Coliseo Gral. Miguel Iturralde de la Universidad de las Fuerzas Armadas – ESPE.** Ubicado dentro de las instalaciones de la Universidad de las Fuerzas Armadas – ESPE, es un espacio interior cerrado y con buena iluminación en el cual se han realizado las pruebas previa autorización del DEPARTAMENTO DE DESARROLLO FÍSICO.
- **Camino exterior fuera de los Hangares del (CICTE).** Ubicado dentro de las instalaciones de la Universidad de las Fuerzas Armadas – ESPE, es un espacio

exterior con influencias del viento, cambios de iluminación por el ambiente y otros.

Los factores que influyen en gran medida al correcto desempeño del aterrizaje automático están relacionados con la detección visual del target y su seguimiento físico, en la Tabla 15 se muestra los factores de interés que influyen para ambos casos en cada uno de los escenarios de prueba.

Tabla 15
Características de interés en los escenarios de pruebas experimentales.

Escenario de Pruebas	Tipo de Iluminación	Cantidad de Iluminación	Intervención de Agentes Externos	Características locales del piso
Hangares del CICTE	Mixta (Natural y Artificial)	Media (Depende del Ambiente)	Media (viento leve)	Pocas (Suelo semi-liso)
Coliseo Gral. Miguel Iturralde	Artificial	Alta	Nula	Regulares (Suelo liso con cambios de intensidad)
Patio Exterior	Natural	Depende del Ambiente	Alta (viento exterior)	Altas (Suelo irregular)

Para iniciar con la experimentación se ha planteado una secuencia de prueba para el aterrizaje (Tabla 16), la secuencia es efectuada en cada uno de los escenarios de prueba.

Tabla 16
Secuencia de acciones para cada escenario de pruebas experimentales.

Secuencia de acciones para las pruebas experimentales	
1. Despegue	Despegar el dron y situarlo a la altura correcta mediante un control manual.
2. Posicionamiento	Control manual para situar al dron sobre el target.
3. Detección inicial y seguimiento	Empezar algoritmo de detección y seguimiento del target por el dron y mantenerlo durante un tiempo definido.
4. Descenso	Cambio del algoritmo al modo de aterrizaje con velocidad de descenso constante.
5. Aterrizaje final	Cuando el dron se encuentra en el umbral de aterrizaje seguro, el comando Land() es enviado por el programa.

Las acciones manuales de la Tabla 16 son realizadas por teclado con el programa para manipulación del controlador “Bebop_autonomy” de la sección 5.1.

6.1 Pruebas experimentales dentro de los Hangares del CICTE

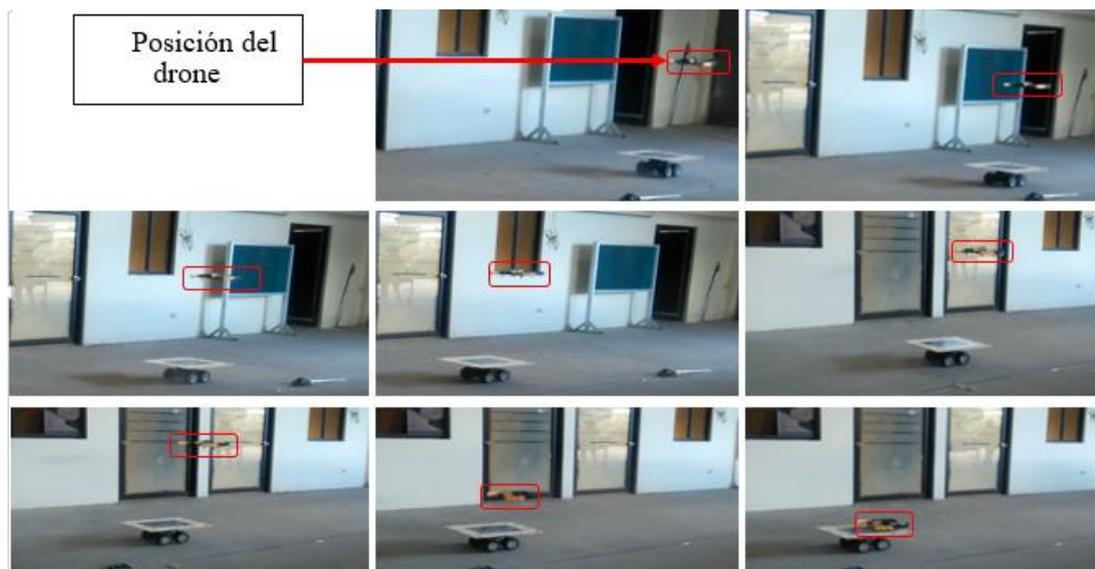


Figura 84. Secuencia de imágenes correspondientes a las pruebas experimentales en los Hangares del CICTE

Para cumplir con los dos campos planteados anteriormente; de detección y seguimiento visual y de control de movimiento, se han realizado seis secuencias de aterrizaje para diferentes condiciones de altura de detección y rapidez de la plataforma terrestre. Cabe destacar que debido a los cambios de iluminación del día en algunas pruebas se ha utilizado una imagen del target directamente de la publicidad para mejor visualización.

6.1.1 Desempeño del seguimiento visual

Para evaluar el desempeño de la detección y seguimiento visual de la plataforma móvil que contiene al target se han tomado en consideración la cantidad de puntos detectados a las diferentes alturas y su correspondencia con los puntos del target.

Las mediciones que tienen validez son las realizadas desde el momento en que se detecta por primera vez al target hasta su aterrizaje en la plataforma móvil. La velocidad de la plataforma terrestre para las secuencias de aterrizaje 1, 2 y 3 de la Tabla 17 es de 0.5 m/s.

Tabla 17
Evaluación de la detección y seguimiento visual para tres secuencias de aterrizaje a 0.5 m/s.

Métrica	Secuencia 1	Secuencia 2	Secuencia 3
Altura de detección	1 metro	1.3 metros	1.5 metros
#Puntos escena	~797.69	~791.07	~781.72
#Correspondencias	~504.37	~486.64	479.03
Tasa de Refrescamientos correctos	1	0.9	0.85

Como se puede observar en la Tabla 17 la altura de detección influye directamente en el desempeño de los algoritmos de extracción de características, no obstante, el algoritmo presentado para la caracterización entre fotogramas consecutivos permite un gran desempeño al utilizar la totalidad de los puntos en la escena para determina el seguimiento visual en lugar de solamente los del target.

Como se esperaba la altura afecta en mayor proporción al refrescamiento de la posición del target, a pesar de ello para las alturas de medición de la Tabla 17 la cantidad de correspondencias sigue siendo correcta dado que el umbral de confianza de detección del target está en los 100 puntos, además al momento de pérdida puede solventarse de la posición relativa brindada por el algoritmo de caracterización entre fotogramas consecutivos.

Las detección también se ve influenciada por la velocidad a la que se realiza el seguimiento, por esta razón se ha realizado otro conjunto de pruebas con una velocidad superior, de igual manera las mediciones que tienen validez son las realizadas desde el momento en que se detecta por primera vez al target hasta su aterrizaje en la plataforma móvil, la velocidad de la misma para las secuencias de aterrizaje 4, 5 y 6 de la Tabla 18 es de 0.7 m/s.

Tabla 18

Evaluación de la detección y seguimiento visual para tres secuencias de aterrizaje a 0.7 m/s.

Métrica	Secuencia 4	Secuencia 5	Secuencia 6
Altura de detección	1.1 metros	1.3 metros	1.5 metros
#Puntos escena	~770.09	~776.86	~734.59
#Correspondencias	484.32	~470.66	~434.06
Tasa de Refrescamientos correctos	1	0.89	0.86

Como se evidencia en la Tabla 18 las mediciones a una velocidad superior provocan variaciones negativas en cada uno de los valores de eficiencia para la detección y seguimiento visual, esto es debido a que los cambios de velocidad mayores en el seguimiento disminuyen la calidad en la captura de imágenes.

A pesar de la reducción de eficiencia en los algoritmos de detección de puntos de interés debido a las diferentes alturas y velocidades, se puede evidenciar que el número de correspondencias sigue superando el umbral mínimo estimado para una buena

detección en el capítulo IV que es de 100 puntos para refrescamiento entre fotogramas consecutivos y 70 puntos para refrescamiento del target.

6.1.2 Desempeño del control de movimientos

La evaluación de desempeño del control de movimientos, por otra parte, se ha realizado con la disminución del error de posición respecto al target mediante el seguimiento del valor del “Set Point” generado, como se explicó en el capítulo anterior, el set point es una función proporcional al error de posición del centro de masa del target con respecto a la proyección del centro óptico de la cámara.

Para una mejor visualización en esta prueba se mostrarán imágenes comparativas para una sola de las secuencias, posteriormente se mostraran los resultados acumulados, en la Figura 85 se muestra el seguimiento del set point por la medición de los movimientos estimados entre fotogramas consecutivos durante todo el proceso de aterrizaje únicamente para la secuencia 6.

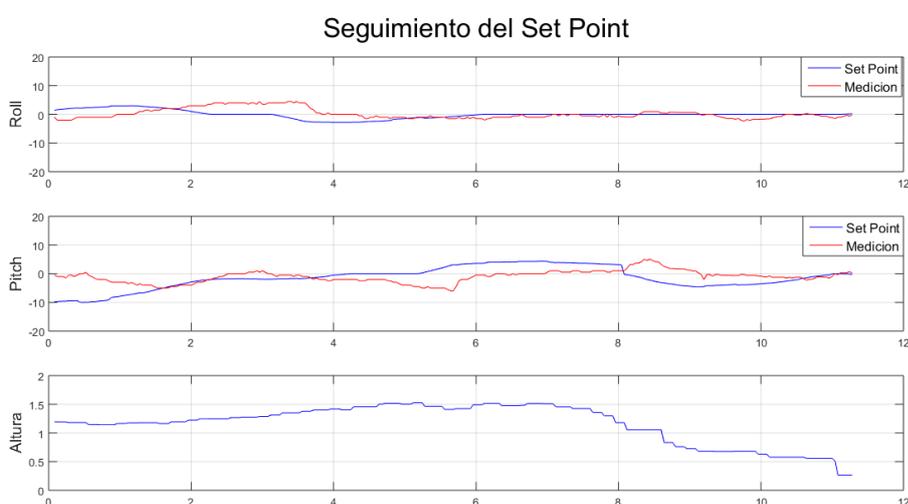


Figura 85. Medición del desplazamiento generado con respecto al deseado en el tiempo en el proceso de aterrizaje para la secuencia 6.

Como se puede observar en la Figura 85 la medición del desplazamiento tiende a acercarse al set point que va cambiando en función del error de posición respecto al target. A pesar de los cambios en el set point el acercamiento sucede en un tiempo menor a 4 segundos de manera que al llegar a la etapa final del algoritmo se logra un error considerablemente pequeño.

La Figura 86 presenta el error de desplazamiento, de los movimientos generados de la Figura 85. Cabe mencionar que cada medición es relativa en distancia respecto a la escala ofrecida por la Altura que se observa en la parte inferior de la Figura 86.

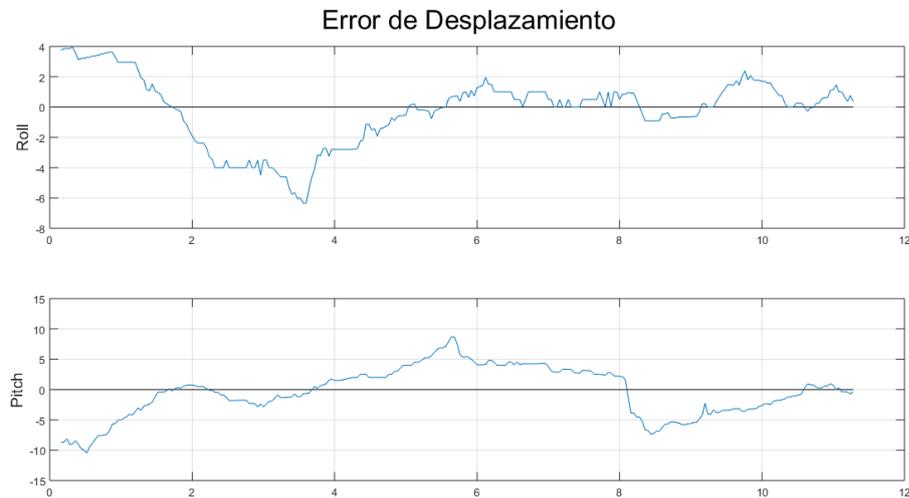


Figura 86. Error de desplazamiento en el control del drone para la secuencia 6.

Como se observa en la Figura 86 el error de desplazamiento para la secuencia 6 va disminuyendo hasta llegar al punto de aterrizaje para ambos ejes en donde el error es menor a 0.5 de la medida esperada.

Finalmente la distancia en metros desde el drone hasta el target en los ejes X, Y para la secuencia 6 se puede observar en la Figura 87, donde se evidencia la disminución del error de distancia entre el drone y el target hasta llegar al punto donde se aterriza finalmente.

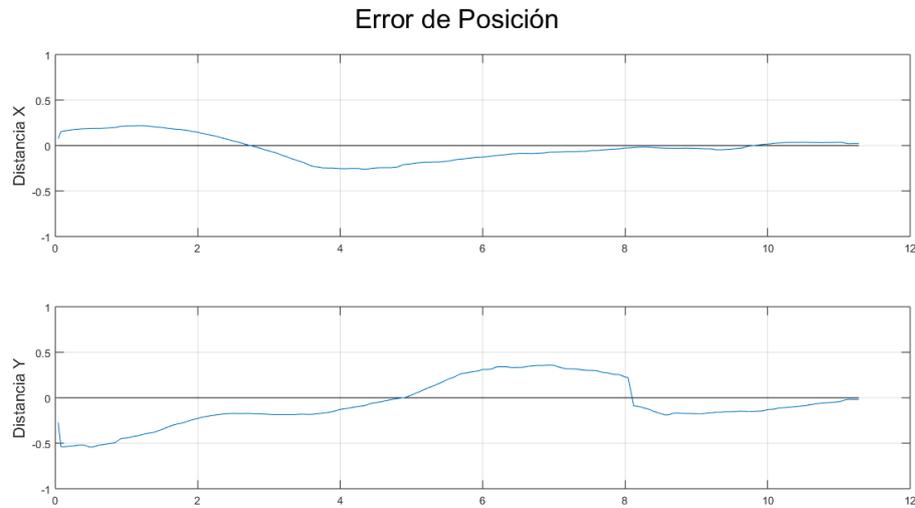


Figura 87. Error de posición en metros desde el drone hasta el target en los ejes X, Y para la secuencia 6.

El aterrizaje para la secuencia 6 se realizó con un error de menos de cinco centímetros en ambos ejes como se puede observar en la Figura 87.

Finalmente la Figura 88 y Figura 89 presentan las mediciones de error en el desplazamiento para cada instante de tiempo y el error de posición para todas las secuencias de las pruebas dentro del hangar.

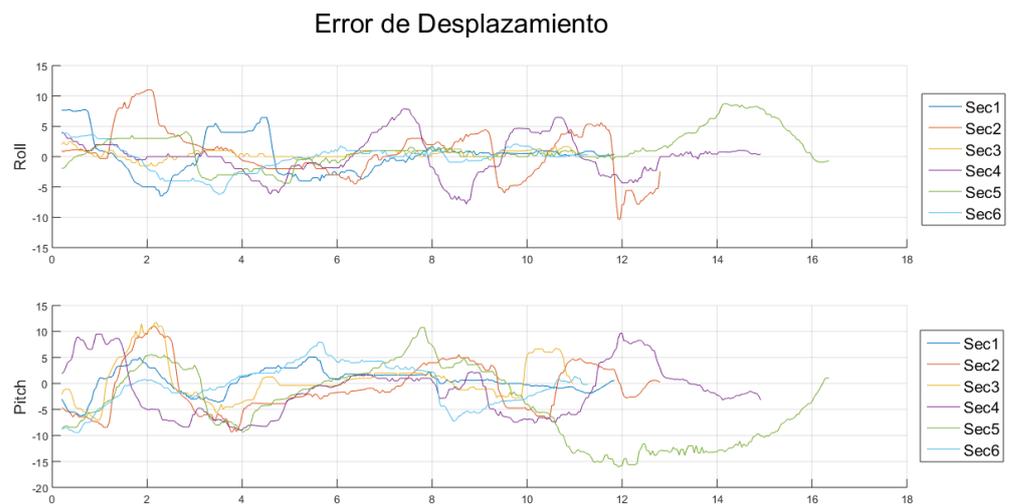


Figura 88. Error de desplazamiento en cada instante de tiempo para las secuencias de prueba en el hangar.

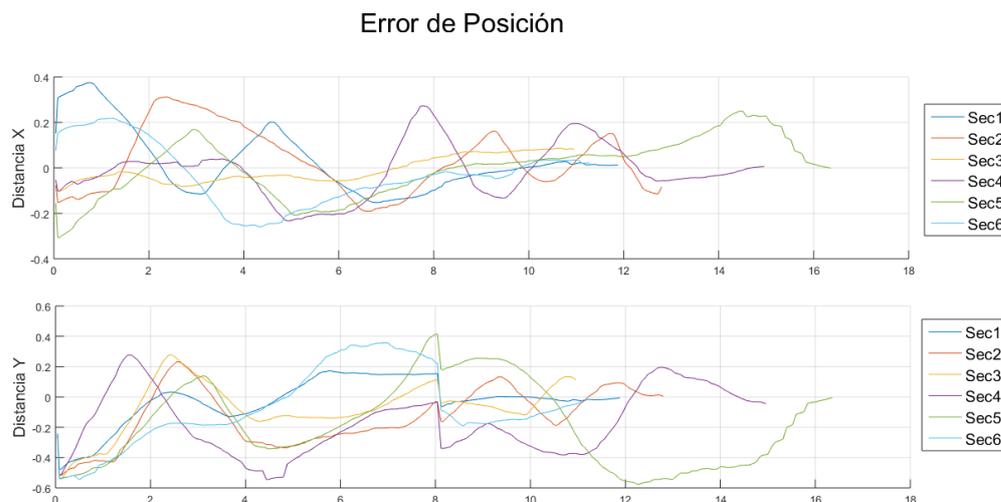


Figura 89. Error de posición en metros desde el drone hasta el target en los ejes X, Y para las secuencias de prueba en el hangar.

La posición medida en la que aterrizó finalmente el drone en cada una de las secuencias se presenta en la Tabla 19.

Tabla 19
Error de posición (cm) al momento del aterrizaje para las secuencias de prueba en el hangar.

Secuencia	Error visual X	Error visual Y	Error medido X	Error medido Y	Lugar de aterrizaje
1	1.252	-0.46	5	-8	Dentro
2	-8.329	0.43	-14	5	Dentro
3	8.162	10.9	-6	22	Parcialmente fuera
4	0.5678	4.45	7	3	Dentro
5	-0.0356	0.42	5	3	Dentro
6	2.203	1.91	15	26	Parcialmente Fuera

Las mediciones de la Tabla 19 determinan la distancia desde el punto de seguimiento hasta el centro óptico de visión, es decir la cámara del drone en (cm). A continuación se presentan las imágenes tomadas un instante antes del aterrizaje, después de haber enviado la instrucción “Land()” donde se puede evidenciar la

posición del dron antes de aterrizar Figura 90, en algunas de las pruebas el dron quedo parcialmente colgado de la plataforma.

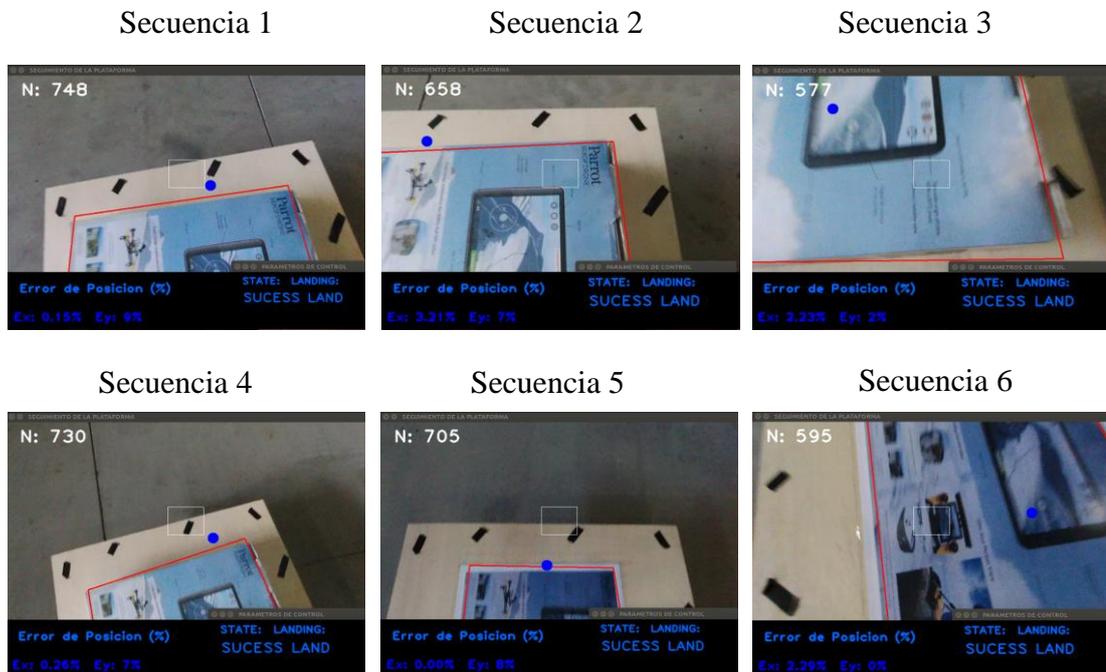


Figura 90. Imágenes tomadas antes del aterrizaje para las secuencias de prueba en el hangar.

6.2 Pruebas experimentales dentro del Coliseo Gral. Miguel Iturralde

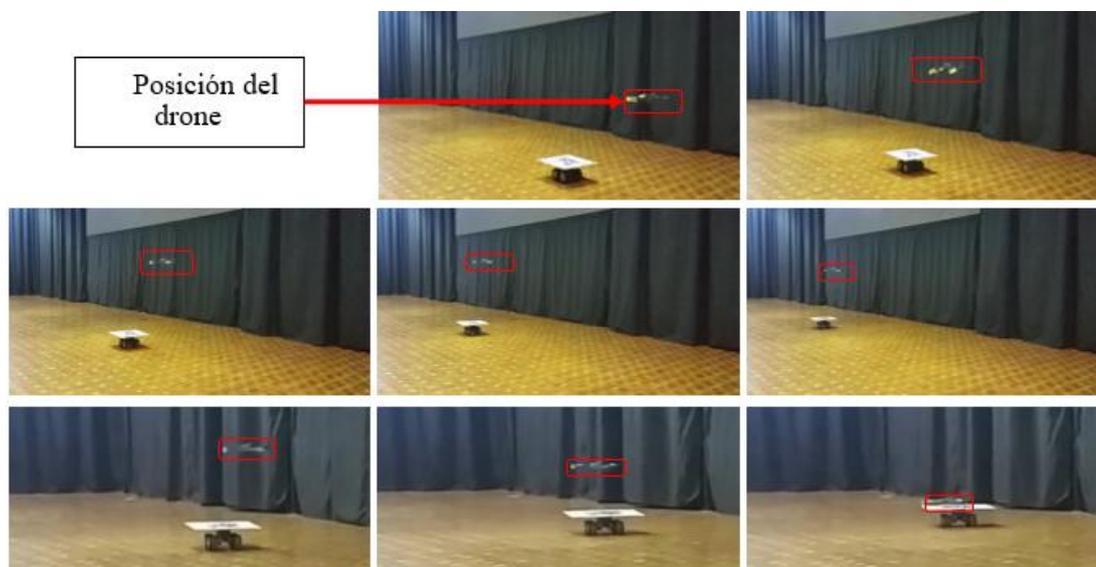


Figura 91. Secuencia de imágenes correspondientes a las pruebas experimentales en el Coliseo Gral. Miguel Iturralde.

En el Coliseo Gral. Miguel Iturralde la iluminación artificial es claramente aprovechable para no tener problemas de mala visualización del target. Se han realizado un conjunto de cinco secuencias de aterrizaje a diferentes alturas para identificar el rendimiento de los algoritmos en un entorno bien iluminado.

Se han tomado en cuenta las mismas condiciones del anterior conjunto de pruebas, con la variación de que en esta prueba se evalúa el algoritmo para la velocidad de 0.8 m/s de la plataforma terrestre en todas las secuencias. La prueba final en el coliseo se realizó a la altura de 2 metros para lo cual fue necesario usar la imagen del target directamente de la publicidad para mejor visualización del algoritmo.

6.2.1 Desempeño del seguimiento visual

En las secuencias de aterrizaje correspondientes a esta prueba se tienen mejores condiciones de visualización por la iluminación artificial dentro del coliseo, para evaluar el desempeño bajo estas condiciones se ha tomado en cuenta la cantidad de puntos detectados por el algoritmo y sus correspondencias correctas con el target. Las mediciones que tienen validez son las realizadas desde el momento en que se detecta por primera vez al target hasta su aterrizaje en la plataforma móvil.

Tabla 20

Evaluación del seguimiento visual para cinco secuencias de aterrizaje a 0.8 m/s de la plataforma terrestre dentro del coliseo.

	Altura de detección	#Puntos escena	#Correspondencias	Tasa de Refrescamientos correctos
Secuencia 1	1.2 metros	790.45	395.32	0.89
Secuencia 2	1.4 metros	795.104	410.57	0.78
Secuencia 3	1.6 metros	788.78	419.19	0.65
Secuencia 4	1.8 metros	793.71	452.92	0.5
Secuencia 5	2 metros	793.75	435.10	0.74

Como se puede apreciar en la Tabla 20 la captación de características no refleja deficiencias considerables a los cambios de altura en el algoritmo que empareja correspondencias entre fotogramas consecutivos, pero se puede apreciar que la tasa de refrescamientos si se ve afectada en gran medida ya que los cambios de altura escalan al target a niveles superiores a los apreciables por el umbral del algoritmo haciendo del refrescamiento más complicado por la cantidad disminuida de puntos correspondientes al mismo encontrados en la escena. A pesar de los inconvenientes el algoritmo de emparejamiento entre fotogramas consecutivos logra solventar el seguimiento visual durante el tiempo en el cual no se realizan refrescamientos para posteriormente aterrizar el drone.

6.2.2 Desempeño del control de movimientos

Para la evaluación de desempeño del control de movimientos en el coliseo se han comparado los efectos de disminución del error de posición respecto al target mediante el seguimiento del valor del “Set Point” generado.

En la Figura 92 se evidencia el seguimiento del set point por la medición de los movimientos generados entre fotogramas consecutivos durante todo el proceso de aterrizaje para la secuencia 2.

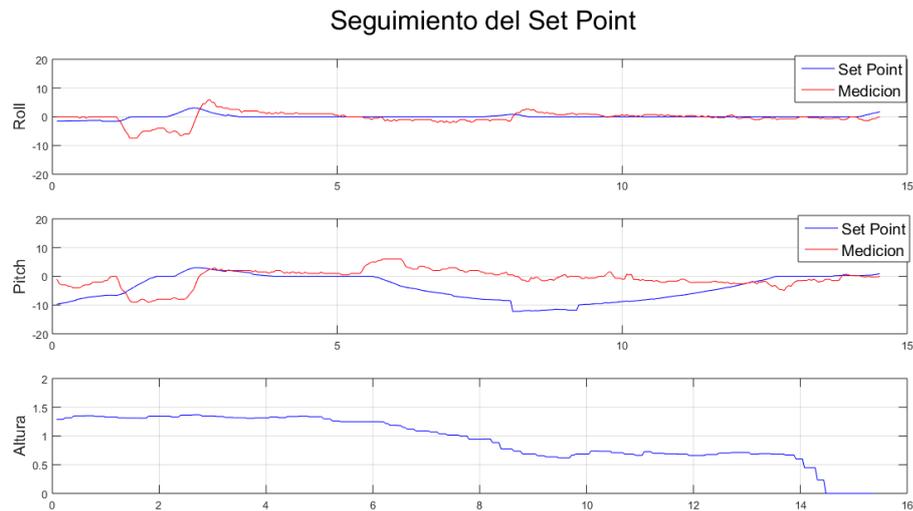


Figura 92. Medición del desplazamiento generado con respecto al deseado en el tiempo en el proceso de aterrizaje para la secuencia 2.

Para medir el desempeño del aterrizaje la Figura 93 muestra el error en el desplazamiento obtenido en contraste con el deseado (set point) para cada una de las secuencias en esta prueba.

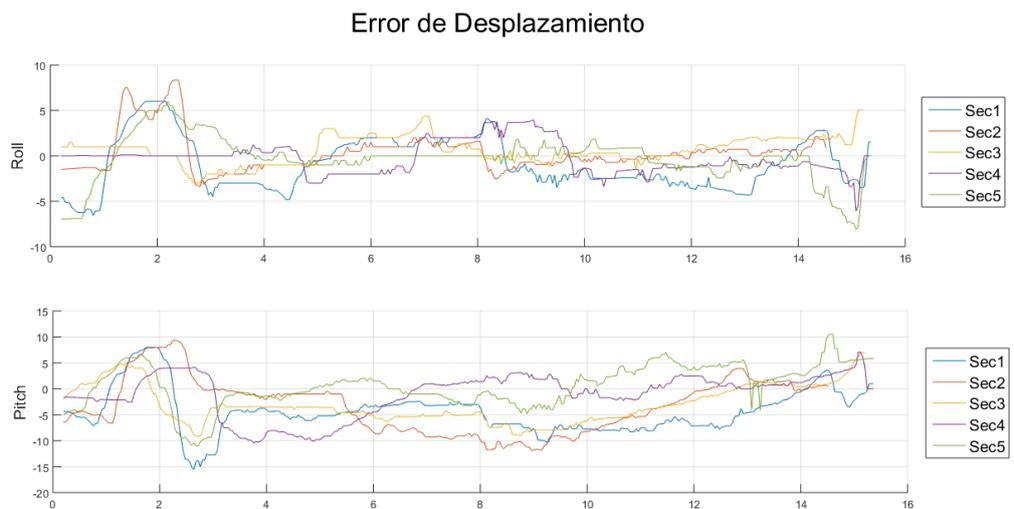


Figura 93. Error de desplazamiento en cada instante de tiempo para las secuencias de prueba en el coliseo.

Como se aprecia en la Figura 93 el error en todos los casos llega a un punto de inflexión cerca del final de la rutina que es cuando se envía la instrucción final de aterrizaje, después de ello los errores son producto de no poder realizar movimientos dado que el drone se queda quieto en su punto de aterrizaje, para todos los casos el final de la rutina presenta errores de hasta siete unidades en desplazamiento al aterrizar. En la Figura 94 se puede ver la distancia en metros desde el drone hasta el target en los ejes X, Y para todas las secuencias de prueba en el coliseo, donde se evidencia la disminución del error de distancia entre el drone y el target hasta llegar al punto donde se aterriza finalmente.

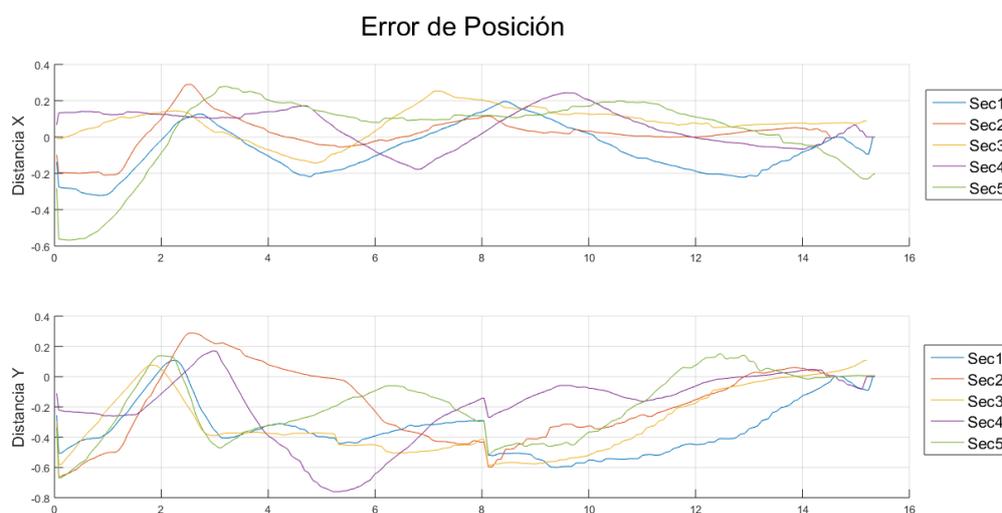


Figura 94. Error de posición en metros desde el drone hasta el target en los ejes X, Y para las secuencias de prueba en el coliseo.

En la Figura 94 se puede apreciar como la distancia decrece en gran medida hasta llegar al punto de aterrizaje, en las primeras cuatro secuencias la posición final de aterrizaje no supera los 15 cm desde el target hasta el centro óptico proyectado de la cámara, demostrando el aterrizaje dentro de la plataforma que mide $60 \times 60 \text{ cm}$. En la secuencia final la altura de detección causó problemas haciendo que el aterrizaje falle. En la Tabla 21 se puede apreciar el error de posición al final del aterrizaje en cada una de las secuencias.

Tabla 21

Error de posición (cm) al momento del aterrizaje para las secuencias de prueba en el coliseo.

Secuencia	Error visual X	Error visual Y	Error medido X	Error medido Y	Lugar de aterrizaje
1	0.124	-2.14	16	-2	Dentro
2	-2.286	1.135	14	7	Dentro
3	-14.382	-15.641	-15	4	Dentro
4	12.890	-9.619	18	-16	Parcialmente fuera
5	-19.407	3.56	-14	30	Fuera

Como se muestra en la Tabla 21 dos de las secuencias tienen problemas al aterrizar, esto se presenta debido a la altura de detección que ocasionó cambios bruscos en el punto de seguimiento al final de la rutina. En la Figura 95 se pueden observar las imágenes captadas un instante antes de aterrizar, después de haber enviado el comando “Land()”.

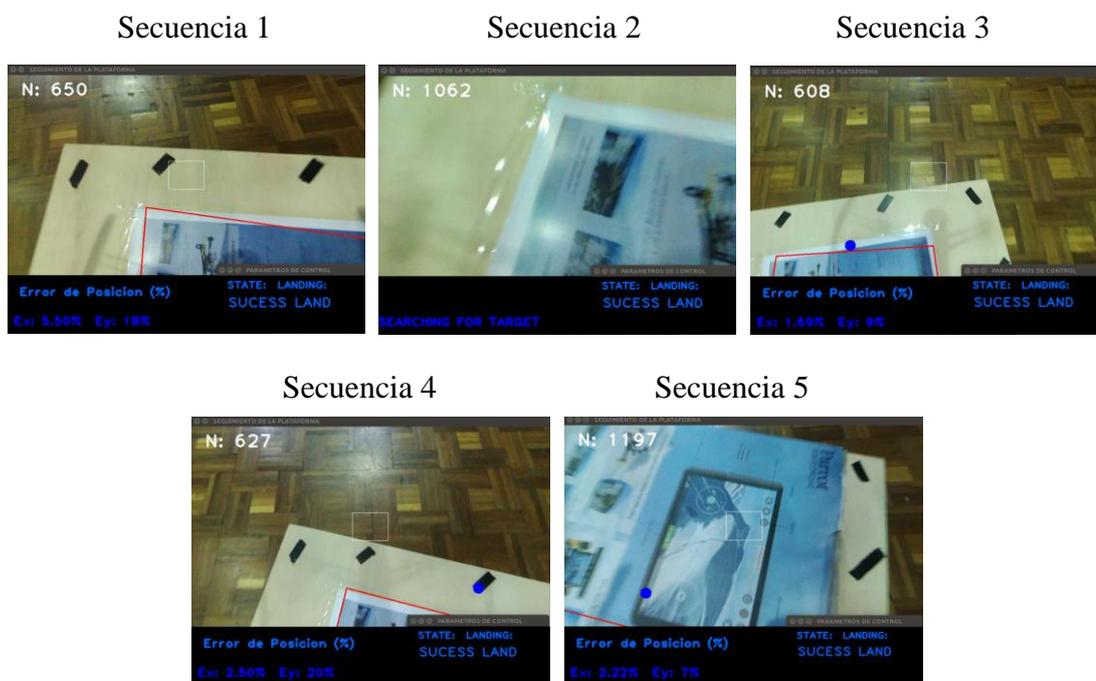


Figura 95. Imágenes tomadas antes del aterrizaje para las secuencias de prueba en el coliseo.

6.3 Pruebas experimentales en el patio exterior al Hangar del CICTE

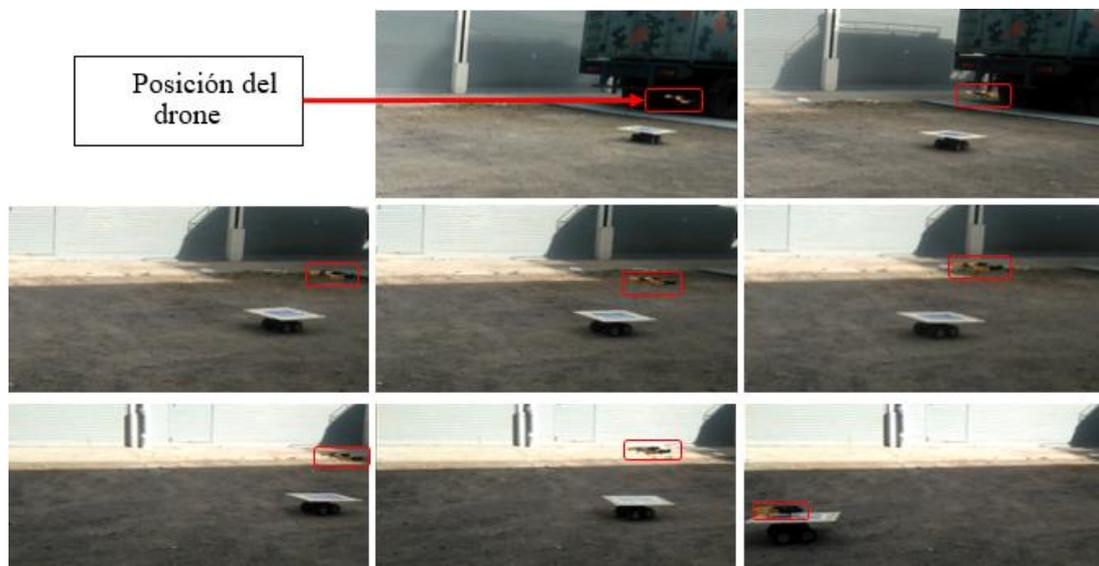


Figura 96. Secuencia de imágenes correspondientes a las pruebas experimentales en el patio exterior a los hangares del CICTE.

En el patio exterior a los hangares del CICTE las condiciones ambientales son las regulares de un espacio abierto con interferencias del viento y cambios de luz por ser iluminación natural, a continuación se presenta el análisis de desempeño para la misma rutina de aterrizaje automático de las pruebas anteriores en un conjunto de tres secuencias con velocidad de la plataforma móvil de 0.6 m/s. Cabe destacar que debido a la iluminación cambiante del exterior en algunas de las pruebas se ha utilizado como target la imagen original directa de la publicidad.

6.3.1 Desempeño del seguimiento visual

El desempeño del seguimiento visual para estas condiciones tiende a ser influenciado por las perturbaciones debido al viento y cambios de iluminación del día, tomando en cuenta únicamente las mediciones desde el momento de la detección hasta el aterrizaje final se obtienen los resultados de la Tabla 22.

Tabla 22

Evaluación del seguimiento visual para tres secuencias de aterrizaje a 0.6 m/s de la plataforma terrestre en el patio exterior al Hangar del CICTE.

	Altura de detección	#Puntos escena	#Correspondencias	Tasa de Refrescamientos correctos
Secuencia 1	1.2 metros	796.14	495.26	0.97
Secuencia 2	1.4 metros	798.29	451.92	0.85
Secuencia 3	1.6 metros	768.89	448.33	0.62

Como se puede observar la escena exterior tampoco presenta grandes impactos desfavorables en la detección y emparejamiento de características entre fotogramas consecutivos, pero la tasa de refrescamientos al igual que en pruebas anteriores decrece con el escalamiento debido a la altura de detección.

6.3.2 Desempeño del control de movimientos

Para el seguimiento del set point en esta prueba se ha tomado la secuencia 1 que presenta errores menores a cinco unidades al final de su rutina, el error más importante es el tiempo de acercamiento que se evidencia en la Figura 97 pues para pitch no se logra un valor estable en las cercanías del set point debido a las perturbaciones ocasionadas por el viento.

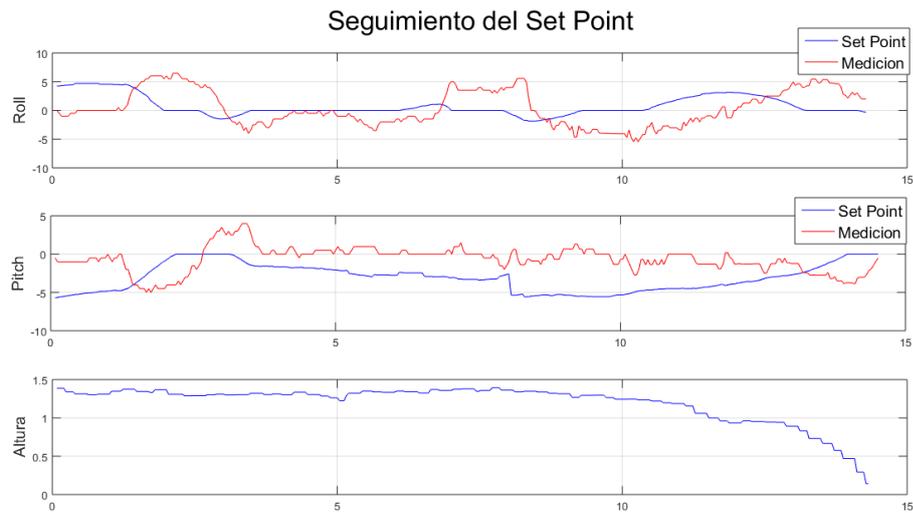


Figura 97. Medición del desplazamiento generado con respecto al deseado en el tiempo en el proceso de aterrizaje para la secuencia 1.

Los errores de desplazamiento de las mediciones estimadas con respecto al set point generado para las tres secuencias de aterrizaje se presentan en la Figura 98.

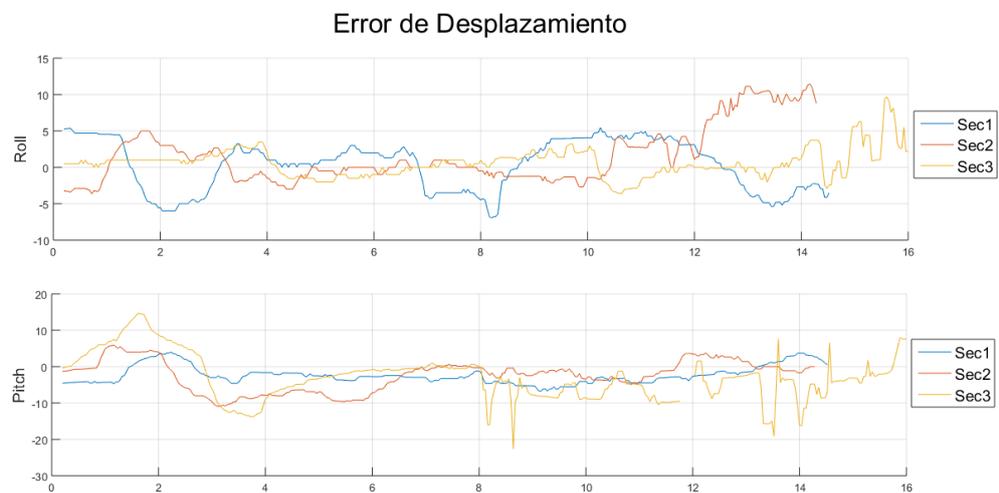


Figura 98. Error de desplazamiento en cada instante de tiempo para las tres secuencias exteriores.

Como se puede observar en la Figura 98 el error de desplazamiento en el caso del control en Roll se ve influenciado al final para terminar con errores de hasta nueve pixeles que representan alrededor de 2 cm por cada interacción. A continuación se

presenta la distancia en metros desde el drone hasta el target en los ejes X, Y para todas las secuencias de prueba.

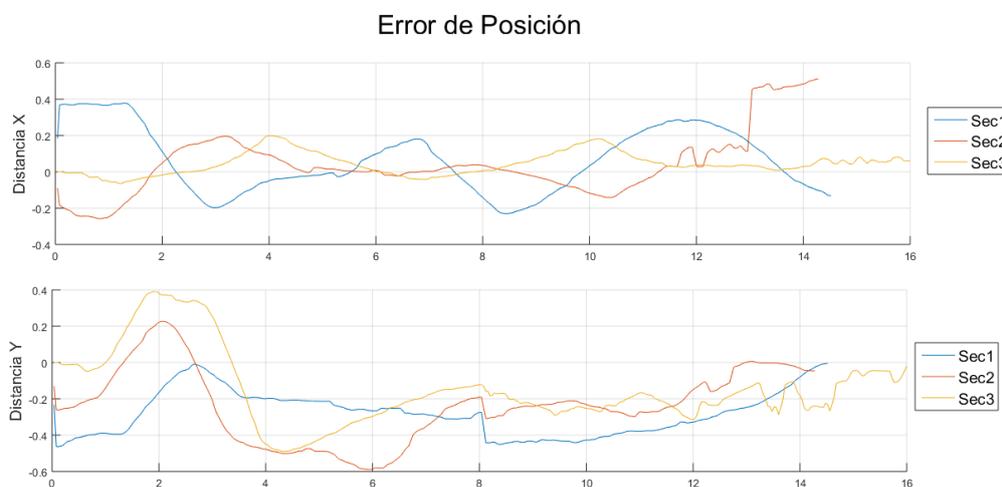


Figura 99. Error de posición en metros desde el drone hasta el target en los ejes X, Y para las secuencias de prueba exterior.

Como se puede apreciar en la Figura 99 el aterrizaje es forzado a pesar de no disminuir al mínimo la distancia entre el target y el drone ya que la influencia del viento cambia la respuesta del control haciéndolo más lento. Al final dos de las pruebas han sido exitosas con valores de error en el eje de las abscisas de entre 15 cm y en las ordenadas no supera los cinco centímetros ratificando el aterrizaje final dentro de la plataforma móvil por sus dimensiones de 60×60 cm. No obstante en la segunda prueba los movimientos bruscos han causado fallas en la detección y el error final. En la Tabla 23 se puede evidenciar el error de posición en (cm) para cada una de las secuencias.

Tabla 23

Error de posición (cm) al momento del aterrizaje para las secuencias de prueba en el hangar.

Secuencia	Error visual X	Error visual Y	Error medido X	Error medido Y	Lugar de aterrizaje
1	-13.15	-1.92	-28	5	Parcialmente fuera
2	51.11	-4.62	53	12	Fuera
3	6.12	-2.03	13	7	Dentro

Finalmente las imágenes de la Figura 100 han sido tomadas un instante antes del aterrizaje, después de haber enviado el comando “Land()” y demuestran la posición del drone antes del aterrizaje final.



Figura 100. Imágenes tomadas antes del aterrizaje para las secuencias de prueba en el patio fuera del coliseo.

CAPÍTULO VII

CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones

En la etapa de caracterización de imágenes se realizó una comparación del desempeño de tres de los algoritmos más representativos en la literatura SIFT, SURF y ORB en un conjunto de imágenes y secuencias de video definidos que contienen al target, obteniendo un desempeño notablemente superior de ORB por su rapidez en el proceso y calidad equiparable con SURF.

Se realizó una evaluación de la capacidad máxima de extracción de características con respecto al tiempo dándole en promedio a ORB un 85.2% de efectividad con respecto a los otros dos algoritmos, además se evaluó la calidad de la caracterización de puntos de interés con el concepto de recall, precisión y tiempo para cada uno de los algoritmos obteniendo una medida en promedio de recall superior en SURF de 0.317 contra ORB con un 0.3096, no obstante en precisión y tiempo ORB fue superior a ambos algoritmos con el 59.4% de precisión y un tiempo promedio de caracterización de 0.021 segundos.

Durante la etapa de seguimiento visual del target y aplicando el algoritmo ORB en la caracterización de fotogramas consecutivos, se obtuvo mejores resultados en la detección continua del target. Esto se debe a los cambios mínimos inter-fotograma y un mayor número de puntos de interés.

Se logró reducir los efectos de deformaciones y cambios geométricos erróneos en el recuadro que representa la posición del target, producto del error acumulado en el algoritmo de caracterización entre fotogramas consecutivos, con la implementación de un refrescamiento periódico de su posición mediante la caracterización del target original a la porción de ventana estimada que lo contiene. Con dicho complemento al algoritmo de caracterización de fotogramas consecutivos logra un mejor posicionamiento del target y disminución de las incertidumbres en la detección visual.

La estimación del modelo de movimiento basado en el procesamiento de las imágenes capturadas a 25 fps logró solventar los problemas de compartición de información de estado del dron que se limita 5 Hz. La estimación se realizó para movimientos en los ejes correspondientes al plano paralelo al de tierra, es decir movimientos en Pitch y Roll.

El uso de la transformación afín para estimar los parámetros de movimiento entre fotogramas consecutivos en el modelado del dron permitió una buena aproximación a los desplazamientos en el plano paralelo al piso debido a los movimientos rectilíneos ejercidos en la rutina correspondiente y las características planas del piso en el coliseo. Los datos que se capturaron con acciones de control de la velocidad lineal superiores a 0.5 y 0.6 en Roll y Pitch respectivamente fueron descartados debido a que presentaron superposición de los valores de medidas de control más bajas por la distorsión de la imagen a velocidades muy altas.

El uso de un controlador PID solventó la persecución de la plataforma móvil mediante el seguimiento de un delta de desplazamiento proporcional al error de posición del dron con respecto al target. La posición del target es resultado de la estimación del movimiento entre fotogramas consecutivos.

La utilización de la altura como escala asociada al movimiento logró complementar al modelo de movimiento obtenido por estimación con transformación afín, de manera que el seguimiento y aterrizaje del dron se lograron mediante el error de posición estimado con base a la altura de detección.

Finalmente se realizaron pruebas experimentales de funcionamiento para evaluar el desempeño del sistema en tres diferentes escenarios bajo diferentes condiciones de visualización y perturbaciones del ambiente. Las pruebas se realizaron en los Hangares del Centro de Investigación Científica y Tecnológica del Ejército (CICTE), en el Coliseo Gral. Miguel Iturralde y en el Patio exterior a los hangares del CICTE.

De las pruebas realizadas dentro de los hangares del CICTE se puede concluir que el cambio de altura en la detección influye desfavorablemente en el desempeño del aterrizaje dado que las pruebas a 1.5 metros terminaron en el aterrizaje del dron parcialmente fuera de la plataforma móvil.

Las pruebas realizadas en el coliseo permitieron llegar a la altura de 2 metros en el seguimiento gracias a la iluminación superior brindada en el lugar, donde se evidenció que al aumentar la altura de detección el refrescamiento de la posición del target pierde efectividad comprometiendo el seguimiento de la plataforma, y a pesar que el algoritmo de caracterización entre fotogramas consecutivos logra una estimación de la posición constante, las pruebas a 1.8 y 2 metros terminaron comprometidas cayendo parcialmente y fuera de la plataforma móvil en el aterrizaje.

De las pruebas de funcionamiento al aire se pudo concluir que a pesar de haber tenido una tasa de refrescamientos en su mayoría buenos con un índice de 0.62 y mayores la acción del viento ralentiza el control de movimientos del drone ocasionando errores en el aterrizaje final con mediciones de hasta 53 cm de diferencia en la posición respectiva al punto de seguimiento en el target.

Hay que mencionar que las pruebas han sido realizadas para una plataforma relativamente pequeña y a diferentes condiciones de visualización, aun así los resultados son satisfactorios en la mayoría de casos y el sistema presenta un buen funcionamiento a pesar de las condiciones ambientales. Además el sistema funciona en tiempo real gracias a la utilización de puntos de interés para la localización y seguimiento de la plataforma móvil dándole la factibilidad de ser implementado “on board” en futuras aplicaciones.

7.2 Recomendaciones

Se recomienda la utilización del controlador de ROS “bebop_autonomy” para comunicación con el drone Bebop de Parrot, ya que ofrece la posibilidad de acceder a todas las funcionalidades del drone mediante tópicos, servicios y parámetros.

Se recomienda tener precauciones al momento de volar el drone, los vuelos interiores pueden ser peligrosos dado que se puede perder la comunicación por interferencias u otros imprevistos como choques, para ello se recomienda utilizar las protecciones EPP para evitar accidentes en vuelos internos.

La duración de las baterías del Bebop drone 1, es de aproximadamente 10 minutos en vuelo, no obstante el procesador que incluye el drone es muy potente y suele

recalentarse con la utilización de una sola batería por ello se recomienda que los vuelos se realicen en lapsos separados de tiempo para evitar mal funcionamiento del equipo.

Se recomienda aprovechar la información visual y las propiedades de cámara virtual de la cámara de 180° del Bebop para realizar el modelamiento “off board” debido a las limitaciones en la entrega de información de estado que dispone el dron.

El sistema de aterrizaje automático está basado en visión artificial y por tanto es susceptible a cambios bruscos de visualización, se recomienda no realizar maniobras evasivas cuando el dron solo logra ver un pequeño espacio dado que la distorsión de las imágenes puede ocasionar transformaciones geométricas erróneas en el recuadro del target dando como resultado errores en las acciones de control del dron.

Se recomienda la utilización de un objetivo que no coincida en su totalidad con las formas o figuras presentes en la escena dado que la caracterización podría presentar falsos positivos indiscriminables.

Se recomienda además no superar los límites de velocidad mencionados en este trabajo para las órdenes de control en Pitch y Roll ya que una vez superado ese límite la caracterización presenta errores debido a las distorsiones de la imagen por las altas velocidades.

REFERENCIAS BIBLIOGRÁFICAS

- Achille, C., Adami, A., Chiarini, S., Cremonesi, S., Fassi, F., Fregonese, L., & Taffurelli, L. (2015). UAV-based photogrammetry and integrated technologies for architectural applications—methodological strategies for the after-quake survey of vertical structures in Mantua (Italy). *Sensors (Switzerland)*, *15*(7), 15520-15539.
- Aguilar, W. G., & Angulo, C. (2012). Compensación de los Efectos Generados en la Imagen por el Control de Navegación del Robot Aibo ERS 7. En *Memorias del VII Congreso de Ciencia y Tecnología ESPE 2012* (pp. 165-170).
- Aguilar, W. G., & Angulo, C. (2014). Estabilización de vídeo en micro vehículos aéreos y su aplicación en la detección de caras. En *IX Congreso de Ciencia y Tecnología ESPE 2014* (pp. 155-160).
- Aguilar, W. G., & Angulo, C. (2014). Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, *2014*(1), 46.
- Aguilar, W. G., & Angulo, C. (2014). Robust Video Stabilization based on Motion Intention for low-cost Micro Aerial Vehicles. En *Systems Signals and Devices (SSD), 2014 11th International Multi-Conference on* (pp. 1-6). IEEE.
- Aguilar, W. G., & Angulo, C. (2016). Real-time model-based video stabilization for micro aerial vehicles. *Neural processing letters*, *43*(2), 459-477.
- Aguilar, W. G., & Angulo Bahón, C. (2013). Estabilización robusta de vídeo basada en diferencia de nivel de gris. En *Memorias del VIII Congreso de Ciencia y Tecnología*. Sangolquí.
- Aguilar, W. G., Casalglla, V. P., & Polit, J. L. (2017). Obstacle Avoidance for Low-Cost UAVs. En *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on* (pp. 503-508). IEEE.
- Aguilar, W. G., Casalglla, V., & Pólit, J. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, *6*(1), 10.

- Aguilar, W. G., Casaliglla, V., Pólit, J., Abad, V., & Ruiz, H. (2017). Obstacle avoidance for flight safety on unmanned aerial vehicles. En *International Work-Conference on Artificial Neural Networks* (pp. 575-584). Springer, Cham.
- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Parra, H., & Ruiz, H. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers with Meanshift. En *IEEE 11th International Conference on Semantic Computing, ICSC 2017* (pp. 509-514). IEEE.
- Aguilar, W. G., Luna, M., Moya, J., Abad, V., Ruiz, H., Parra, H., & Angulo, H. (2017). Pedestrian detection for UAVs using cascade classifiers and saliency maps. En *International Work-Conference on Artificial Neural Networks* (pp. 563-574). Springer, Cham.
- Aguilar, W. G., Luna, M., Moya, J., Abad, V., Ruiz, H., Parra, H., & Lopez, W. (2017). Cascade Classifiers and Saliency Maps Based People Detection. En *International Conference on Augmented Reality, Virtual Reality and Computer* (pp. 501-510). Springer, Cham.
- Aguilar, W. G., & Morales, S. (2016). 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. *Electronics*, 5(4), 70.
- Aguilar, W. G., Morales, S., Ruiz, H., & Abad, V. (2017). RRT* GL based optimal path planning for real-time navigation of UAVs. En *International Work-Conference on Artificial Neural Networks* (pp. 585-595). Springer, Cham.
- Aguilar, W. G., Morales, S., Ruiz, H., & Abad, V. (2017). RRT* GL Based Path Planning for Virtual Aerial Navigation. En *International Conference on Augmented Reality, Virtual Reality and Computer* (pp. 176-184). Springer, Cham.
- Aguilar, W. G., Rodríguez, G. A., & Álvarez, L. (2017). On-Board Visual SLAM on a UGV Using a RGB-D Camera. En *International Conference on Intelligent Robotics and Applications* (Vol. 8102, pp. 298-308). Springer, Cham.
- Aguilar, W. G., Rodríguez, G., Álvarez, L., Sandoval, S., & Quisaguano, F. (2017). Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing. En

- International Conference on Augmented Reality, Virtual Reality and Computer* (pp. 410-419). Springer, Cham.
- Aguilar, W. G., Rodríguez, G., Álvarez, L., Sandoval, S., & Quisaguano, F. (2017). Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing. En *International Work-Conference on Artificial Neural Networks* (pp. 596-606). Springer, Cham.
- Aguilar, W. G., Salcedo, V. S., Sandoval, D. S., & Cobeña, B. (2017). Developing of a Video-Based Model for UAV Autonomous Navigation. En B. C. Barone D., Teles E. (Ed.), *Latin American Workshop on Computational Neuroscience. Communications in Computer and Information Science* (Vol. 720, pp. 94-105). Springer, Cham.
- Azinheira, J. R., & Rives, P. (2008). Image-based visual servoing for vanishing features and ground lines tracking: Application to a UAV automatic landing. *International Journal of Optomechatronics*, 2(3), 275-295.
- Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3), 346-359.
- Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded up robust features. En *European Conference on Computer Vision* (Vol. 3951 LNCS, pp. 404-417). Springer, Berlin, Heidelberg.
- Berrutti, J., Falkenstein, L., & Favaro, F. (2015). *Vuelo autónomo de un cuadricóptero*. Universidad de la República.
- Bohdanov, D. (2012). *Quadrotor UAV Control for Vision-based Moving Target Tracking Task. Tesis Doctoral*. University of Toronto (Canadá).
- Charmette, B., Royer, E., & Chausse, F. (2016). Vision-based robot localization based on the efficient matching of planar features. *Machine Vision and Applications*, 27(4), 415-436.
- Cheng-Yi, Chen and Shih-Hong, C. (2012). Improved SIFT algorithm to match points in the texture region image. En *33rd Asian Conference On Remote Sensing*.

- Chmaj, G., & Selvaraj, H. (2015). Distributed Processing Applications for UAV/drones: A Survey. En *Progress in Systems Engineering. Advances in Intelligent Systems and Computing* (Vol. 366, pp. 449-454). Springer, Cham.
- Clarke, R. (2014). Understanding the drone epidemic. *Computer Law and Security Review*, 30(3), 230-246.
- Custers, B. (2016). Drones Here, There and Everywhere Introduction and Overview. En *The Future of Drone Use* (Vol. 27, pp. 3-20). T.M.C. Asser Press, The Hague.
- Duffy, J. P., Cunliffe, A. M., DeBell, L., Sandbrook, C., Wich, S. A., Shutler, J. D., ... Anderson, K. (2017). Location, location, location: considerations when using lightweight drones in challenging environments. *Remote Sensing in Ecology and Conservation*, 1-13.
- Engel, J., Sturm, J., & Cremers, D. (2012a). Accurate Figure Flying with a Quadcopter Using Onboard Visual and Inertial Sensing. *Workshop proceedings of the International Conference on Intelligent Robot Systems*, 2815–2821. Recuperado a partir de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6385458%5Cnhttps://vision.in.tum.de/_media/spezial/bib/engel12vicomor.pdf
- Engel, J., Sturm, J., & Cremers, D. (2012b). Camera-based navigation of a low-cost quadcopter. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2815-2821. Recuperado a partir de <http://ieeexplore.ieee.org/document/6385458/>
- Engel, J., Sturm, J., & Cremers, D. (2014). Scale-Aware Navigation of a Low-Cost Quadcopter with a Monocular Camera. *Robotics and Autonomous Systems*, 62(11), 1646-1656.
- Enrique, M., & Fernández, G. (2016). *Desarrollo de un sistema de aterrizaje autónomo sobre una plataforma de carga para un UAV tipo Parrot AR.Drone 2*. Escuela Técnica Superior de Ingenieros Industriales – UPM.
- Fan, B., Wang, Z., & Wu, F. (2015). *Local Image Descriptor: Modern Approaches*. Springer Berlin Heidelberg.

- Forsyth, D., & Ponce, J. (2012). *Computer vision: a modern approach*. Pearson.
- González, I. (2003). *Estudio experimental de métodos de calibración y autocalibración de cámaras (Tesis Doctoral)*. Universidad de las plamas de Gran Canaria.
- Grauman, K., & Leibe, B. (2011). Visual Object Recognition. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(2), 1-181.
- Guenard, N., Hamel, T., & Mahony, R. (2007). A practical visual servo control for an unmanned aerial vehicle. *IEEE Transactions on Robotics*, 24(2), 331-340.
- Hartley, R., & Zisserman, A. (2003). *Multiple View Geometry in Computer Vision* (Second edi). Cambridge university press.
- Hassaballah, M., Abdelmgeid, A. A., & Alshazly, H. A. (2016). Image Feature Detectors and Descriptors. En *Image Feature Detectors and Descriptors* (Vol. 630, pp. 11-45). Springer, Cham.
- Hassanalain, M., & Abdelkefi, A. (2017). Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91, 99-131.
- Ilbay, M. Ñ., Córdova, L. T., Raura, G., & Gualotuña, T. (2013). *Análisis De Eficiencia En Algoritmos De Reconocimiento De Imágenes Digitales Aplicables a Dispositivos Móviles Bajo La Plataforma Android (Bachelor's thesis)*. Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Sistemas e Informática.
- Işık, Ş., & Özkan, K. (2014). A Comparative Evaluation of Well-known Feature Detectors and Descriptors. *International Journal of Applied Mathematics, Electronics and Computers*, 3(1), 1-6.
- Jáuregui, M., Hernández, Domingo, Ospino, Héctor, Chacón, & Leira. (2015). Generación de modelos 3D a partir de dos imágenes fotogramétricas 3D models generation from two photogrammetric images. *Ciencia e Ingeniería.*, 37(1), 11-17.
- Juan, L., & Gwun, O. (2009). A comparison of sift, pca-sift and surf. *International*

Journal of Image Processing (IJIP), 3(4), 143-152.

Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc.

Kendoul, F. (2012). Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems. *Journal of Field Robotics*, 29(2), 315-378.

Krig, S. (2014). Interest Point Detector and Feature Descriptor Survey. En *Computer Vision Metrics* (pp. 217-282). Apress, Berkeley, CA.

Kühling, C. (2017). *Fisheye Camera System Calibration for Automotive Applications*. Freie Universität Berlin.

Leos Monroy, G. (2014). *Implementación de una aeronave no tripulada de despegue vertical de cuatro rotores*. Universidad Nacional Autónoma de México.

Limnaios, G. (2014). Current Usage of Unmanned Aircraft Systems (UAS) and Future Challenges: A Mission Oriented Simulator for UAS as a Tool for Design and Performance Evaluation. *Journal of Computations & Modelling*, 4(1), 167-188.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. En *The Proceedings of the Seventh IEEE International Conference on Computer Vision* (Vol. 2, pp. 1150-1157). IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91-110.

Ma, Y., Kosecka, J., Soatto, S., & Sastry, S. (2003). *An Invitation to 3-D Vision: From Images to Models*. Springer Verlag.

Mani Monajjemi (AutonomyLab, S. F. U. (2015). bebop_autonomy - ROS Driver for Parrot Bebop Drone (quadrocopter) 1.0 & 2.0. Recuperado 10 de mayo de 2017, a partir de <http://bebop-autonomy.readthedocs.io/en/latest/index.html>

Maza, I., Caballero, F., Capitán, J., Martínez-De-Dios, J. R., & Ollero, A. (2011). Experimental results in multi-UAV coordination for disaster management and civil security applications. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 61(1-4), 563-585.

- Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1615–1630.
- Nho, K., & Agarwal, R. K. (2000). Automatic Landing System Design Using Fuzzy Logic, 23(2), 298-304.
- Nixon, M. S., & Aguado, A. S. (2012). *Feature Extraction & Image Processing for Computer Vision*. Academic Press.
- OpenCV homepage. (s. f.). Recuperado 10 de enero de 2018, a partir de <https://opencv.org/>
- Ortiz González, J. P. (2014). *Modelado y control robusto de un vehículo aéreo no tripulado quadrotor, en ambientes cerrados (Tesis de Maestría)*. Universidad Politécnica Salesiana.
- Papakonstantinou, A., Topouzelis, K., & Pavlogeorgatos, G. (2016). Coastline Zones Identification and 3D Coastal Mapping Using UAV Spatial Data. *ISPRS International Journal of Geo-Information*, 5(6), 75.
- Parra Muñoz, M., Feitosa, E., & Alves da Silva, J. (2013). Modelamiento matemático y control de un helicóptero de cuatro motores Mathematical modeling and control of a four-engine helicopter. *Scientia et Technica*, 18(4).
- Parrot gallery. (s. f.). Recuperado 10 de enero de 2018, a partir de <http://global.parrot.com/usa/gallery/bebop-drone/?page=2>
- Platero, C. (2009). Formación de imágenes. En *Apuntes de Visión Artificial* (pp. 33-89). Recuperado a partir de http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap2VAFormImagv1.pdf
- Quan, Q. (2017). *Introduction to Multicopter Design and Control*. Singapore: Springer Singapore.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... Mg, A. (2009). ROS: an open-source Robot Operating System. En *ICRA workshop on open*

source software (Vol. 3, No. 3.2, p. 5).

Quigley, M., Gerkey, B., & Smart, W. D. (2015). *Programming Robots with ROS: a practical introduction to the Robot Operating System*. O'Reilly Media, Inc.

ROS (Robot Operating System). (s. f.). Recuperado 10 de enero de 2018, a partir de <http://www.ros.org>

Rosten, E., & Drummond, T. (2005). Fusing points and lines for high-performance tracking. En *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (pp. 1508-1515). IEEE.

Rosten, E., & Drummond, T. (2006). Machine Learning for High Speed Corner Detection. En *European conference on computer vision* (Vol. 1, pp. 430-443). Springer, Berlin, Heidelberg.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. En *Computer Vision (ICCV), 2011 IEEE international conference on* (pp. 2564-2571). IEEE.

Saber, A., & Sayem, S. (2016). *Vision-Aided Navigation for Autonomous Vehicles Using Tracked Feature Points (Tesis de Maestría)*. EMBRY-RIDDLE Aeronautical University.

Serrano, D. (2011). Introduction to ROS -Robot Operating System-, 1-12. Recuperado a partir de <http://pub1.willowgarage.com/~konolige/cs225B/docs/quigley-icra2009-ros.pdf>

Singh, S., & Padhi, R. (2009). Automatic path planning and control design for autonomous landing of UAVs using dynamic inversion. En *American Control Conference, 2009. ACC '09* (pp. 2409-2414). IEEE.

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1), 1-21.

Tuytelaars, T., & Mikolajczyk, K. (2008). Local Invariant Feature Detectors: A Survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3), 177-

280. Recuperado a partir de <http://www.nowpublishers.com/article/Details/CGV-017>

Ubuntu homepage. (2018). Recuperado 10 de enero de 2018, a partir de <https://www.ubuntu.com/>

Urzua, S., Munguía, R., & Grau, A. (2017). Vision-based SLAM system for MAVs in GPS-denied environments. *International Journal of Micro Air Vehicles*, 9(4), 283-296.

Viala, C. R. (2008). *Catacterización y optimización del proceso de calibrado de cámaras basado en Plantilla bidimensional (Tesis Doctoral)*. Universidad Politécnica de Valencia. <https://doi.org/10.4995/Thesis/10251/1858>

Vyškovský, A. (2014). *Object tracking by a flying drone*. Charles University in Prague Faculty of Mathematics and Physics.

Wenzel, K. E., Masselli, A., & Zell, A. (2011). Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 61(1-4), 221-238.

Whitten, W. D. (2017). *Improving the robustness of monocular vision-aided navigation for multirotors through integrated estimation and guidance (Tesis Doctoral)*.

Zhu, Y., Shen, X., & Chen, H. (2016). Copy-move forgery detection based on scaled ORB. *Multimedia Tools and Applications*, 75(6), 3221-3233.